ROBÓTICA

TERCERA EDICIÓN



JOHN J. CRAIG

Robótica

Tercera edición

John J. Craig

TRADUCCIÓN:
Alfonso Vidal Romero Elizondo
Ingeniero en Sistemas Electrónicos
ITESM - Campus Monterrey

REVISIÓN TÉCNICA:
José Ramón Álvarez Bada
Departamento de Ingeniería Eléctrica y Electrónica
ITESM - Campus Ciudad de México



México • Argentina • Brasil • Colombia • Costa Rica • Chile • Ecuador
España • Guatemala • Panamá • Perú • Puerto Rico • Uruguay • Venezuela

Datos de catalogación bibliográfica

ROBÓTICA

Craig, John J.

PEARSON EDUCACIÓN, México, 2006
ISBN: 970-26-0772-8
Área: Ingeniería

Formato: 18.5 × 23.5 cm Páginas: 408

ISBN 0201543613

Edición en español
Editor: Pablo Miguel Guerrero Rosas
e-mail: pablo.guerrero@pearsoned.com
Editor de desarrollo: M. Bernardino Gutiérrez Hernández
Supervisor de Producción: Rodrigo Romero Villalobos

Edición en inglés
Vice President and Editorial Director, ECS: *Marcia J. Horton*
Associate Editor: *Alice Dworkin*
Editorial Assitant: *Carole Snyder*
Vice President and Director of Production and Manufacturing, ESM: *David W. Riccardi*
Executive Managing Editor: *Vince O'Brien*
Managing Editor: *David A. George*
Production Editor: *James Buckley*
Director of Creative Services: *Paul Belfanti*
Art Director: *Jayne Cante*
Cover Designer: *Bruce Kenselaar*
Art Editor: *Greg Dulles*
Manufacturing Manager: *Trudy Pisciotti*
Manufacturing Buyer: *Lisa McDowell*
Senior Marketing Manager: *Holly Stark*

TERCERA EDICIÓN, 2006

Atlacomulco 500-5o. piso
Col. Industrial Atoto
53519, Naucalpan de Juárez, Edo. de México
E-mail: editorial.universidades©pearsoned.com

Cámara Nacional de la Industria Editorial mexicana. Reg. Núm. 1031

ISBN 970-26-0772-8

Impreso en México. *Printed in Mexico.*
1 2 3 4 5 6 7 8 9 0 - 09 08 07 06

Contenido

Prefacio

Por lo general, los científicos tienen la sensación de que por medio de su trabajo están aprendiendo sobre algún aspecto de ellos mismos. Los físicos ven esta conexión en su trabajo; e igual pasa con los psicólogos y los químicos. En la robótica, la conexión entre el campo de estudio y nosotros mismos es inusualmente obvia. Y, a diferencia de una ciencia que solamente analiza, el objetivo actual de la robótica requiere que la ingeniería se desvíe hacia la síntesis. Tal vez por eso sea tan fascinante para muchos de nosotros el estudio de esta ciencia.

La robótica se relaciona en sí con el deseo de sintetizar algunos aspectos de la función humana mediante el uso de mecanismos, sensores, actuadores y computadoras. Obviamente esto representa un enorme compromiso que evidentemente parece requerir una multitud de ideas provenientes de varios campos "clásicos".

En la actualidad expertos de varios campos trabajan en la investigación de distintos temas de la robótica. Generalmente, no es común que un solo individuo domine todos los campos de la robótica. Es natural esperar una subdivisión. En un nivel relativamente alto de abstracción parece razonable dividir la robótica en cuatro áreas principales: manipulación mecánica, locomoción, visión computacional e inteligencia artificial.

Este libro introduce la ciencia y la ingeniería de la manipulación mecánica. Esta subdisciplina de la robótica tiene su base en varios campos clásicos; y los más relevantes son la mecánica, la teoría de control y la ciencia computacional. Los capítulos del 1 al 8 tratan temas relacionados con la ingeniería mecánica y las matemáticas. Del 9 al 11 cubren todo lo relacionado con la teoría de control, y los capítulos 12 y 13 pueden clasificarse como material relacionado con la ciencia computacional. A pesar de esta división, en todo el libro se enfatizan los aspectos computacionales de los problemas; por ejemplo, cada capítulo relacionado con la mecánica tiene una breve sección dedicada a las consideraciones computacionales.

Este libro evolucionó a partir de los resúmenes utilizados en clase para impartir la materia "Introducción a la Robótica", en la Universidad de Stanford, durante las clases de otoño de 1983 a 1985. Las primeras dos ediciones se utilizaron en muchas instituciones de 1986 hasta 2002. La tercera edición se beneficia de este uso e incorpora correcciones y muchas mejoras gracias a la retroalimentación recibida. Agradecemos a todos aquellos que han enviado correcciones del texto al autor.

El libro es apropiado para un curso a nivel de los últimos dos años de licenciatura o del primer año de maestría. Será de mayor utilidad si el estudiante ha llevado un curso básico de estática y dinámica, uno de álgebra lineal, y si puede programar en un lenguaje de alto nivel. Además es conveniente, aunque no absolutamente necesario, que haya completado un curso introductorio en teoría de control. Uno de los objetivos del libro es presentar el material de una manera simple e intuitiva; no es necesario que la audiencia esté formada estrictamente por ingenieros mecánicos, aunque la mayor parte del material se toma de ese campo. En Stanford, a muchos ingenieros eléctricos, científicos computacionales y matemáticos les pareció bastante legible el libro.

De manera directa, este libro es para que lo utilicen todos aquellos ingenieros que desarrollan sistemas robóticos, pero el material debe verse como una plataforma importante para cualquiera que tenga algo que ver con la robótica. De manera muy parecida a como los desarrolladores de software generalmente estudian, cuando menos, algo de hardware, quienes no estén directamente involucrados con la mecánica y el control de robots deben tener una base similar a la que se ofrece aquí.

El material se adapta fácilmente a un curso de un semestre académico; para impartir el material en un trimestre probablemente se requiera omitir uno o dos capítulos; e incluso a ese ritmo no podrán cubrirse todos los temas con mucha profundidad. En cierta manera el libro está organizado con esto en mente; por ejemplo, la mayoría de los capítulos presenta solamente un enfoque para resolver el problema en cuestión. Uno de los retos al escribir este libro fue tratar de hacer justicia a los temas cubiertos dentro de las restricciones de tiempo que se presentan generalmente al impartir clases. El método empleado para este fin fue considerar solamente el material que afecta directamente al estudio de la manipulación mecánica.

Al final de cada capítulo se presenta un conjunto de ejercicios, donde cada ejercicio tiene asignado un factor de dificultad, indicado entre corchetes después del número del ejercicio. Los niveles de dificultad varían entre [00] y [50], en donde [00] es un problema trivial y [50] es una investigación sin resolver.[1] Desde luego, lo que para una persona pudiera ser difícil, a otra podría resultarle sencillo, por lo que para algunos lectores los niveles parecerán incorrectos en algunos casos; sin embargo, se ha hecho un esfuerzo por estimar la dificultad de los ejercicios.

En la parte final de cada capítulo hay una tarea de programación en la que el estudiante debe aplicar el contenido del capítulo correspondiente a un manipulador planar simple de tres articulaciones. Este manipulador simple es lo suficientemente complejo como para demostrar casi todos los principios de los manipuladores generales sin necesidad de azorar al estudiante con demasiada complejidad. Cada tarea de programación se basa en las anteriores, con lo que al final del curso contará con una biblioteca completa de software de manipulador.

Adicionalmente, en esta tercera edición hemos agregado ejercicios de MATLAB: un total de 12 ejercicios se encuentran asociados con los capítulos del 1 al 9. El profesor Robert I. Williams II de la Universidad de Ohio es el desarrollador de estos ejercicios; estamos profundamente agradecidos con él por esa contribución. Estos ejercicios pueden utilizarse con la Caja de Herramientas de Robótica MATLAB[2] creada por Peter Corke, Director de investigación de la CSIRO en Australia.

El capítulo 1 es una introducción al campo de la robótica. Presenta cierto material básico, unas cuantas ideas fundamentales y la notación adoptada en el libro; además, muestra un avance de los capítulos siguientes.

El capítulo 2 trata acerca de las matemáticas utilizadas para describir posiciones y orientaciones en el espacio tridimensional. Este material es extremadamente importante: por definición, la manipulación mecánica se refiere al desplazamiento de objetos (piezas, herramientas, el robot en sí) en el espacio. Necesitamos formas de describir esas acciones de una manera que sea fácilmente comprensible y lo más intuitiva posible.

---

[1] He adoptado la misma escala que en la obra *The Art of Computer Programming* de D. Knuth (Addison-Wesley).

[2] Para mayor información sobre la Caja de Herramientas de Robótica MATLAB, visite http://www.ict.csiro.au/robotics/ToolBox7.htm.

Los capítulos 3 y 4 presentan la geometría de los manipuladores mecánicos. Introducen la rama de la ingeniería mecánica conocida como cinemática: el estudio del movimiento sin considerar las fuerzas que lo ocasionan. Estos capítulos tratan con la cinemática de los manipuladores, pero restringiendose a los problemas de posicionamiento estático.

El capítulo 5 expande nuestra investigación de la cinemática a las velocidades y fuerzas estáticas.

El capítulo 6 trata por primera vez las fuerzas y los momentos requeridos para producir el movimiento de un manipulador. Éste es el problema de la dinámica de los manipuladores.

El capítulo 7 se relaciona con la descripción de los movimientos del manipulador en términos de trayectorias a través del espacio.

El capítulo 8 ofrece muchos temas relacionados con el diseño mecánico de un manipulador. Por ejemplo, ¿cuántas articulaciones son apropiadas, de qué tipo deben ser y cómo deben ordenarse?

Los capítulos 9 y 10 estudian métodos para controlar un manipulador (generalmente con una computadora digital) de manera que rastree fielmente una trayectoria de posición deseada a través del espacio. El capítulo 9 restringe la atención a los métodos de control lineal, mientras que el capítulo 10 extiende estas consideraciones al ámbito no lineal.

El capítulo 11 cubre el campo del control de la fuerza activa con un manipulador. Es decir, sobre cómo controlar la aplicación de fuerzas mediante el manipulador. Este método de control es importante cuando el manipulador entra en contacto con el entorno que lo rodea; por ejemplo, al lavar una ventana con una esponja.

El capítulo 12 da un vistazo general a los métodos de programación de robots; específicamente los elementos necesarios en un sistema de programación de robots y los problemas específicos asociados con la programación de robots industriales.

El capítulo 13 presenta la simulación fuera de línea y los sistemas de programación; es decir, la última extensión para la interfaz hombre-robot.

Quiero agradecer a todas las personas que han contribuido con este libro. En primer lugar, agradezco a los estudiantes de ME219 de Stanford su tiempo durante los cursos de otoño de 1983 a 1985, quienes sufrieron con los primeros borradores: encontraron muchos errores y brindaron infinidad de sugerencias. El profesor Bernard Roth ha contribuido de muchas maneras, tanto a través de sus críticas constructivas como proporcionándome un entorno para completar la primera edición. En SILMA Inc. disfruté de un ambiente estimulante, además de contar con los recursos que me ayudaron a completar la segunda edición. El Dr. Jeff Kerr escribió el primer borrador del capítulo 8. El profesor Robert L. Williams II contribuyó con los ejercicios de MATLAB que se encuentran al final de cada capítulo, y Peter Corke expandió su caja de Herramientas de Robótica para apoyar el estilo de la notación Denavit–Hartenberg utilizado en este libro. Tengo una deuda con mis anteriores mentores en robótica: Marc Raibert, Carl Ruoff, Tom Binford y Bernard Roth.

Muchas otras personas de Stanford, SILMA, Adept y de otras instituciones también han ayudado de diversas formas; mi agradecimiento a John Mark Agosta, Mike Ali, Lynn Balling, Al Barr, Stephen Boyd, Chuck Buckley, Joel Burdick, Jim Callan, Brian Carlisle, Monique Craig, Subas Desa, Tri Dai Do, Karl Garcia, Ashitava Ghosal, Chris Goad, Ron Goldman, Bill Hamilton, Steve Holland, Peter Jackson, Eric Jacobs, Johann Jäger, Paul James, Jeff Kerr, Oussama Khatib, Jim Kramer, Dave Lowe, Jim Maples, Dave Marimont, Dave Meer, Kent Ohlund, Madhusudan Raghavan, Richard Roy,

Ken Salisbury, Bruce Shimano, Donalda Speight, Bob Tilove, Sandy Wells y Dave Williams.

Los estudiantes de la clase de Robótica de 2002, impartida por el profesor Roth, en Stanford, utilizaron la segunda edición y enviaron muchas observaciones de errores que se corrigieron para esta tercera edición.

Finalmente, deseo agradecer a Tom Robbins de Prentice Hall por su supervisión con la primera edición y ahora de nuevo con esta tercera edición.

J.J.C.

CAPÍTULO 1

Introducción

1.1 ANTECEDENTES

La historia de la automatización industrial está caracterizada por periodos de cambios bruscos en los métodos populares. Ya sea como causa o, tal vez, como un efecto, dichos periodos de cambio en las técnicas de automatización parecen estar estrechamente ligados con la economía mundial. El uso del **robot industrial**, que se identificó como dispositivo único en la década de 1960 [1], junto con los sistemas de diseño asistido por computadora (CAD) y manufactura asistida por computadora (CAM), caracteriza las tendencias más recientes en la automatización del proceso de manufactura. Estas tecnologías están llevando a la automatización industrial hacia otra transición, cuyo alcance se desconoce aún [2].

En los Estados Unidos hubo mucha adopción de equipo de robótica a principios de la década de 1980, a la cual le siguió un breve retraso a finales de esa misma década. A partir de ese momento el mercado ha estado creciendo (figura 1.1), aunque está sujeto a las variaciones económicas, como pasa con todos los mercados.

En la figura 1.2 se muestra el número de robots instalados por año en las principales regiones industriales del mundo. Observe que Japón reporta números de una manera algo distinta a las demás regiones: cuentan como robots algunas máquinas que en otras partes del mundo no se consideran así (se consideran simplemente como "máquinas de fábrica"). Por ende, los números reportados para Japón están algo inflados.

Una de las principales razones del crecimiento en el uso de robots industriales es la reducción en el costo. En la figura 1.3 se indica que, durante la década de 1990, los precios de los robots disminuyeron al mismo tiempo que aumentaron los costos de mano de obra humana. Además, los robots no sólo se están volviendo más baratos, sino también más efectivos: más rápidos, precisos y flexibles. Si factorizamos en los números estos *ajustes de calidad*, el costo por utilizar robots disminuye mucho más rápido que su etiqueta de precio. A medida que los robots se vuelvan más efectivos en cuanto al costo por sus tareas, y a medida que aumente el costo de la mano de obra humana, habrá más trabajos que se conviertan en candidatos para la automatización robótica. Ésta es la tendencia individual más importante que impulsa el crecimiento del mercado de los robots. Una tendencia secundaria es que, haciendo a un lado la economía, a medida que los robots se vuelvan más *capaces*, podrán hacer más y más tareas que serían peligrosas o imposibles de realizar por los trabajadores humanos.

Las aplicaciones que llevan a cabo los robots industriales se están volviendo cada vez más sofisticadas pero aún se da el caso de que, como en el año 2000, aproximadamente el 78% de los robots instalados en los EE.UU. fueron robots para soldadura o manejo de materiales [3].

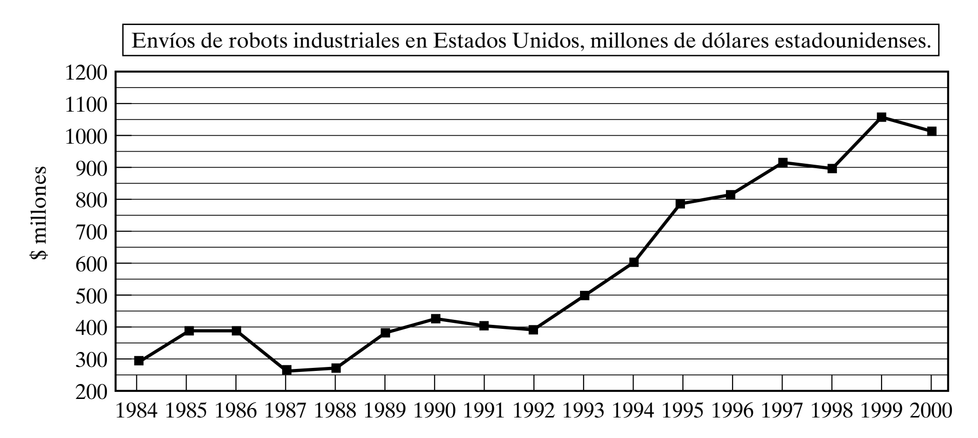


FIGURA 1.1: Envíos de robots industriales en Estados Unidos, en millones de dólares estadounidenses [3].

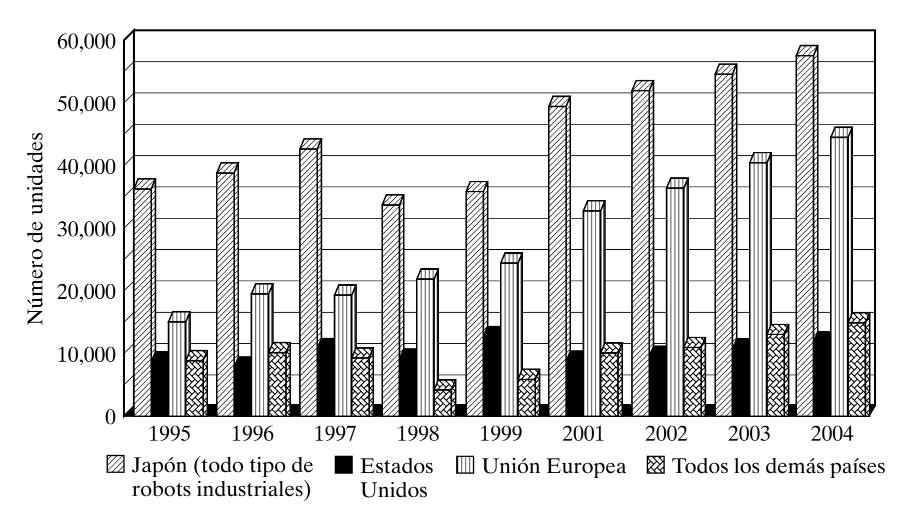


FIGURA 1.2: Instalaciones por año de robots industriales de propósitos múltiples para los años 1995-2000 y las proyectadas para los años 2001-2004 [3].

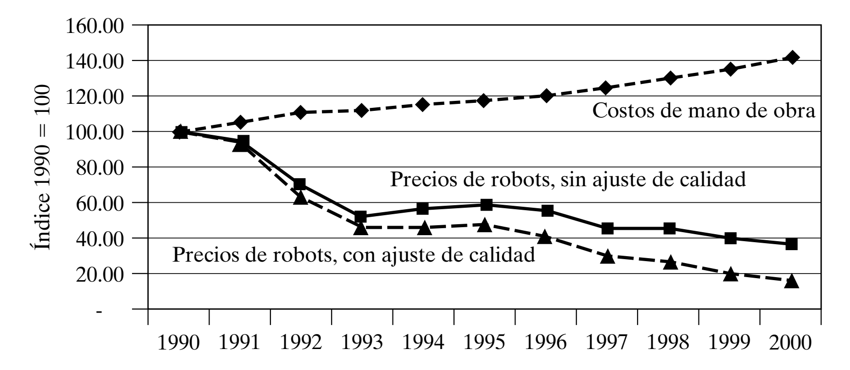


FIGURA 1.3: Precios de los robots comparados con los costos de la mano de obra humana en la década de 1990 [3].



FIGURA 1.4: El manipulador Adept 6 tiene seis articulaciones giratorias y es popular en muchas aplicaciones. Cortesía de Adept Technology, Inc.

Un dominio que ofrece un mayor reto, el **ensamblaje** por un robot industrial, contabilizó el 10% de las instalaciones.

Este libro se enfoca en la mecánica y el control de la forma más importante del robot industrial: el **manipulador mecánico**. Algunas veces se debate sobre qué es lo que constituye exactamente a un robot industrial. Dispositivos como los que se muestran en la figura 1.4 siempre se incluyen, mientras que las máquinas fresadoras controladas numéricamente (NC) generalmente no. La distinción está en alguna parte de la sofisticación de la capacidad de programación del dispositivo; si un dispositivo mecánico puede programarse para realizar una amplia variedad de aplicaciones, probablemente sea un robot industrial. Las máquinas que en su mayor parte están limitadas a una clase de tarea se consideran como de **automatización fija**. Para los fines de este texto, no necesitan debatirse las distinciones; la mayor parte del material es de una naturaleza básica que se aplica a una amplia variedad de máquinas programables.

El estudio de la mecánica y el control de manipuladores no es una ciencia nueva, sino simplemente una colección de temas provenientes de campos “clásicos”. La ingeniería mecánica contribuye con metodologías para el estudio de máquinas en situaciones estáticas y dinámicas. Las matemáticas proveen de herramientas para describir los movimientos espaciales y demás atributos de los manipuladores. La teoría de control proporciona herramientas para diseñar y evaluar algoritmos para realizar los movimientos deseados o las aplicaciones de fuerza. Las técnicas de la ingeniería eléctrica se aplican en el diseño de sensores e interfaces para robots industriales y la ciencia computacional contribuye con la base para programar estos dispositivos para realizar la tarea deseada.

1.2 LA MECÁNICA Y EL CONTROL DE LOS MANIPULADORES MECÁNICOS

En las siguientes secciones presentaremos brevemente cada uno de los temas que cubrirá el texto, así como alguna terminología.

Descripción de posición y orientación

En el estudio de la robótica nos preocupamos constantemente por la ubicación de los objetos en el espacio tridimensional. Estos objetos son los vínculos del manipulador, las piezas y herramientas con las que trabaja y los demás objetos en el entorno del manipulador. A un nivel básico pero importante, estos objetos se describen mediante sólo dos atributos: posición y orientación. Naturalmente, un tema de interés inmediato es la manera en la que representamos estas cantidades y las manipulamos matemáticamente.

Para poder describir la posición y orientación de un cuerpo en el espacio, siempre adjuntamos rígidamente un sistema de coordenadas, o trama, al objeto. Después describimos la posición y orientación de esta trama con respecto a algún sistema de coordenadas de referencia. (Vea la figura 1.5).

Cualquier trama puede servir como sistema de referencia dentro del cual se pueda expresar la posición y orientación de un cuerpo, por lo que a menudo pensamos *en transformar* o *cambiar la descripción de* estos atributos de un cuerpo de una trama a otra. El capítulo 2 explica las convenciones y metodologías para tratar la descripción de la posición y la orientación, y las matemáticas para manipular estas cantidades respecto a varios sistemas de coordenadas.

Es muy útil desarrollar buenas habilidades respecto a la descripción de la posición y la rotación de cuerpos rígidos, incluso en campos externos a la robótica.

Cinemática directa de los manipuladores

La **cinemática** es la ciencia que trata el movimiento sin considerar las fuerzas que lo ocasionan. Dentro de la cinemática se estudian la posición, velocidad, aceleración y todas las derivadas de mayor orden de las variables de posición (respecto al tiempo o a

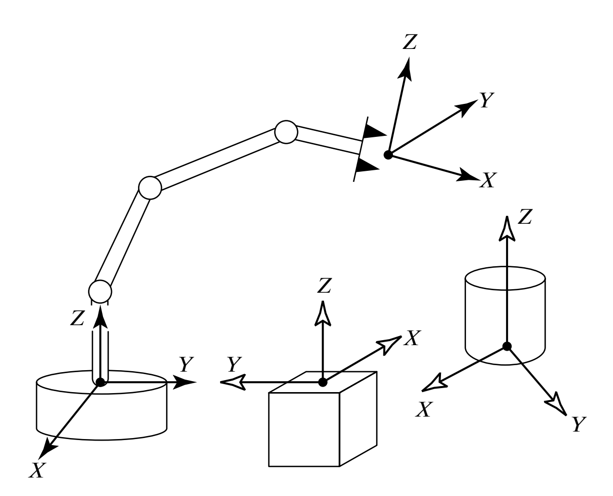


FIGURA 1.5: El sistema de coordenadas o “trama” se adjunta a los manipuladores y a los objetos en el ambiente.

cualquier otra variable). Por ende, el estudio de la cinemática de los manipuladores se refiere a todas las propiedades del movimiento, las geométricas y las basadas en tiempo.

Los manipuladores consisten de **vínculos** casi rígidos, los cuales están conectados por **articulaciones** que permiten el movimiento relativo de los vínculos adyacentes. Estas articulaciones generalmente se instrumentan con sensores de posición, los cuales permiten medir la posición relativa de los vínculos adyacentes. En el caso de las articulaciones giratorias o **angulares**, estos desplazamientos se conocen como **ángulos articulados**. Algunos manipuladores contienen articulaciones deslizantes (o **prismáticas**), en las que el desplazamiento relativo entre los vínculos es una translación, algunas veces llamada **desplazamiento de articulación**.

El número de **grados de libertad** que posee un manipulador es el número de variables de posición independientes que tendrían que especificarse para poder localizar todas las piezas del mecanismo. Éste es un término general que se utiliza para cualquier mecanismo. Por ejemplo, un vínculo de cuatro barras sólo tiene un grado de libertad (incluso aunque haya tres miembros móviles). En el caso de los robots industriales comunes, como un manipulador es generalmente una cadena cinemática abierta y como la posición de cada articulación se define generalmente con una sola variable, el número de articulaciones es igual al número de grados de libertad.

En el extremo libre de la cadena de vínculos que conforman el manipulador se encuentra el **efector final**. Dependiendo de la aplicación que se va a dar al robot, el efector final podría ser una pinza, un soplete de soldadura, un electroimán o cualquier otro dispositivo. Generalmente presentamos la posición del manipulador proporcionando una descripción de la **trama de la herramienta**, la cual está unida al efector final, relativa a la **trama base**, que está unida a la base fija del manipulador. (Vea la figura 1.6).

Un problema muy básico en el estudio de la manipulación mecánica se conoce como **cinemática directa**, que es el problema geométrico estático de calcular la posición y orientación del efector final del manipulador. Específicamente, dado un conjunto de

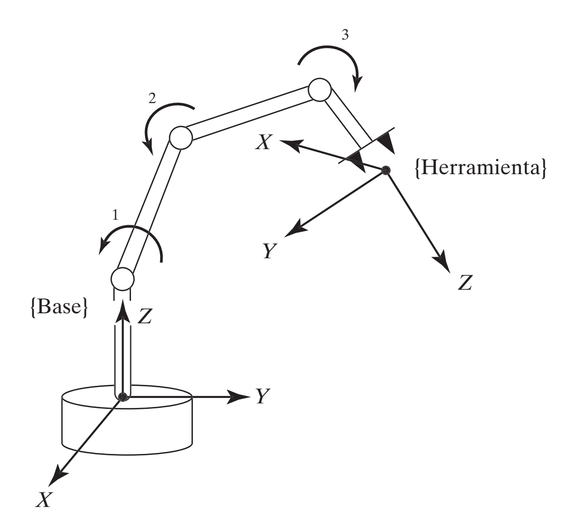


FIGURA 1.6: Las ecuaciones cinemáticas describen la trama de la herramienta relativa a la trama base como una función de las variables de la articulación.

ángulos articulares, el problema de la cinemática directa es calcular la posición y orientación de la trama de la herramienta relativa a la trama base. Imaginemos que es como cambiar la representación de la posición del manipulador: de una descripción en el **espacio de la articulación** a una descripción en el **espacio cartesiano**.[1] Exploraremos este problema en el capítulo 3.

Cinemática inversa de los manipuladores

En el capítulo 4 consideraremos el problema de la cinemática inversa. Este problema se plantea de la siguiente manera: dada la posición y orientación del efector final del manipulador, calcule todos los conjuntos posibles de ángulos articulares que podrían utilizarse para obtener esta posición y orientación dadas. (Vea la figura 1.7). Éste es un problema fundamental en el uso práctico de los manipuladores.

Éste es un problema geométrico algo complicado, que se resuelve de manera rutinaria miles de veces diariamente en el sistema humano y en otros sistemas biológicos. En el caso de un sistema artificial como un robot, necesitamos crear un algoritmo en la computadora de control que pueda realizar este cálculo. En ciertos casos, la solución de este problema es el elemento más importante en un sistema manipulador.

Podemos pensar en este problema como en una *asignación* de “ubicaciones” en el espacio cartesiano 3D, a “ubicaciones” en el espacio de articulaciones internas del robot. Esta necesidad surge naturalmente siempre que se especifica un objetivo en coordenadas de espacio 3D externas. Algunos de los primeros robots carecían de este algoritmo; simplemente se desplazaban (algunas veces manualmente) hacia las ubicaciones deseadas, que después se registraban como un conjunto de valores de articulación (es decir, como una ubicación en el espacio de la articulación) para su posterior reproducción. Obviamente, si el robot se utiliza solamente en el modo de registrar y reproducir las ubicaciones y los movimientos de las articulaciones, no es necesario ningún algoritmo que

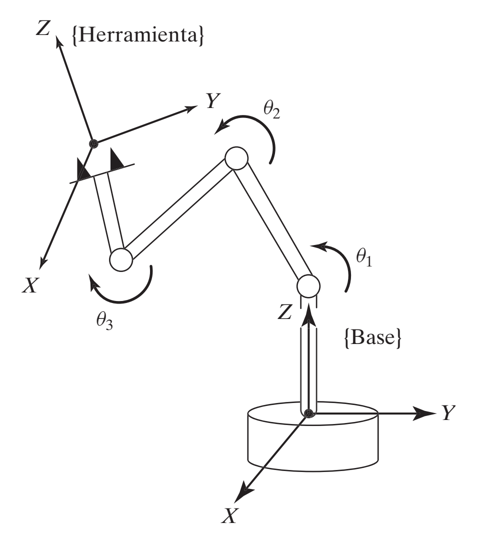


FIGURA 1.7: Para una posición y orientación dadas de la trama de la herramienta, los valores de las variables de una articulación pueden calcularse mediante la cinemática inversa.

[1] Por *espacio cartesiano* nos referimos al espacio en el que la posición de un punto se da mediante tres números, y en donde la orientación de un cuerpo se da también mediante tres números. Algunas veces esto se conoce como *espacio de trabajo* o *espacio operacional*.

relacione el espacio de la articulación con el espacio cartesiano. No obstante, actualmente es raro encontrar un robot industrial que carezca de este algoritmo básico de cinemática inversa.

El problema de la cinemática inversa no es tan simple como el de la cinemática directa. Debido a que las ecuaciones cinemáticas son no lineales, su solución no es siempre sencilla (o incluso posible) en una forma cerrada. Además, surgen preguntas sobre si existe una solución o existen múltiples soluciones.

El estudio de estas cuestiones hace que apreciemos lo que la mente humana y el sistema nervioso logran realizar cuando, al parecer inconscientemente, movemos y manipulamos objetos con nuestros brazos y manos.

La existencia o inexistencia de una solución cinemática define el **espacio de trabajo** de un manipulador dado. La falta de una solución significa que el manipulador no puede obtener la posición y orientación deseadas, ya que se encuentran fuera del espacio de trabajo del manipulador.

Velocidades, fuerzas estáticas, singularidades

Además de tratar con los problemas de posicionamiento estático, es conveniente analizar los manipuladores en movimiento. A menudo, al realizar el análisis de velocidad de un mecanismo, es conveniente definir una matriz conocida como el **jacobiano** del manipulador. El jacobiano especifica la **asignación** de las velocidades en espacio de articulaciones a las velocidades en espacio cartesiano. (Vea la figura 1.8). La naturaleza de esta asignación cambia a medida que varía la configuración del manipulador. En ciertos puntos (llamados **singularidades**), esta asignación no es invertible. Es importante que los diseñadores y usuarios de los manipuladores tengan una comprensión de este fenómeno.

Considere, como ejemplo, el artillero de la parte trasera de un avión de combate biplano de la Primera Guerra Mundial (mostrado en la figura 1.9). Mientras el piloto dirige el avión desde la cabina frontal, el trabajo del artillero en la parte trasera del avión es disparar a las aeronaves enemigas. Para realizar esta tarea, su ametralladora está montada en un mecanismo que gira sobre dos ejes; uno de los movimientos se llama acimut y al otro elevación. Por medio de estos dos movimientos (dos grados de libertad), el artillero puede dirigir su ráfaga de balas en cualquier dirección que desee en el hemisferio superior.

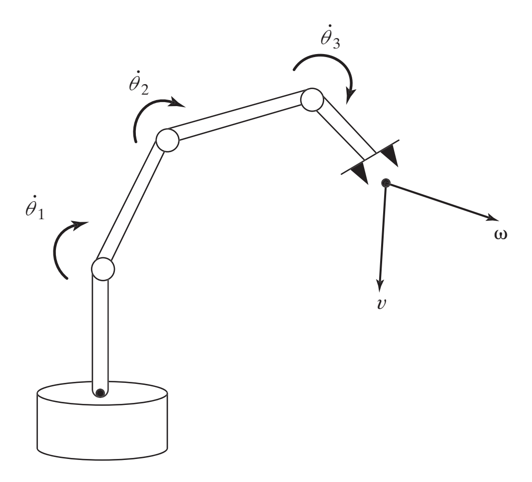


FIGURA 1.8: La relación geométrica entre las proporciones de las articulaciones y la velocidad del efector final pueden describirse en una matriz conocida como el jacobiano.

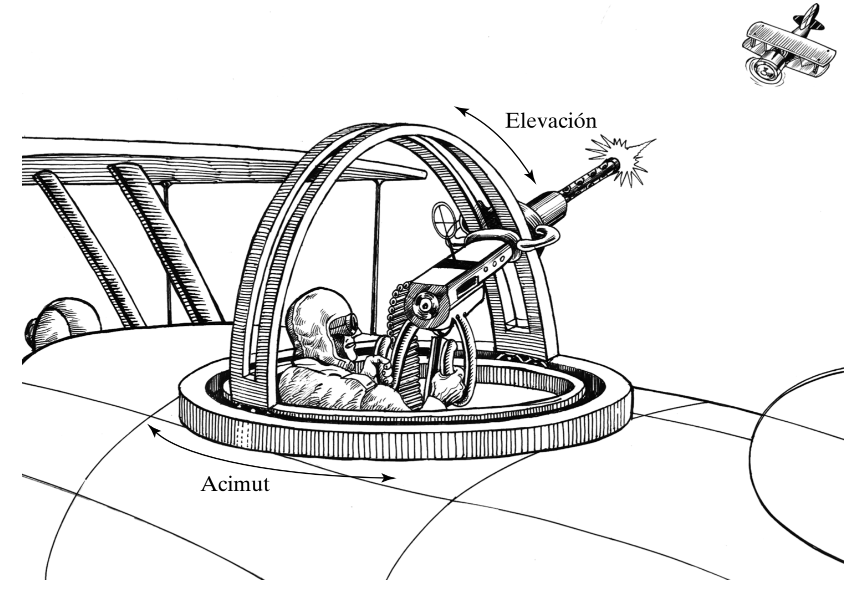


FIGURA 1.9: Un biplano de la Primera Guerra Mundial con un piloto y un artillero en la parte trasera. El mecanismo del artillero en la parte trasera está sujeto al problema de las posiciones singulares.

¡Se detecta un avión enemigo con acimut "a la una en punto" y elevación de 25 grados! El artillero dirige su ráfaga de balas hacia el avión enemigo y rastrea su movimiento para poder pegarle con un flujo continuo de balas el mayor tiempo posible. Logra tener éxito y en consecuencia derriba el avión enemigo.

¡Ahora puede verse un segundo avión enemigo con acimut "a la una en punto" y elevación de 70 grados! El artillero orienta su ametralladora y empieza a disparar. El avión enemigo se mueve para poder obtener una elevación cada vez mayor, relativa al avión del artillero. Pronto el avión enemigo está pasando casi encima. ¿Qué es ésto? ¡El artillero no puede seguir dirigiendo el flujo de balas hacia el avión enemigo! Descubrió que, como el avión enemigo voló por encima, tenía que cambiar su acimut en una proporción muy alta. No pudo deslizar su ametralladora con el acimut requerido lo suficientemente rápido, por lo que ¡el avión enemigo escapó!

En este último escenario, ¡el afortunado piloto enemigo se salvó gracias a una *singularidad*! El mecanismo de orientación de la ametralladora, aún y cuando funciona bien sobre casi todo su rango de operación, se vuelve menos que ideal cuando la ametralladora se dirige casi o directamente hacia arriba. Para rastrear objetivos que pasan a través de la posición directamente encima del avión se requiere de un movimiento muy rápido alrededor del eje de acimut. Entre más cerca pase el objetivo del punto que está directamente encima del avión, más rápido deberá el artillero girar el eje de acimut para rastrear el objetivo. Si el objetivo vuela directamente sobre la cabeza del artillero, ¡tendría que girar la ametralladora en su eje de acimut a una velocidad infinita!

¿Debería quejarse el artillero con el diseñador del mecanismo acerca de este problema? ¿Podría diseñarse un mejor mecanismo? El caso es que el problema realmente no se puede evitar fácilmente. De hecho, cualquier mecanismo de orientación con dos grados de libertad que tenga exactamente dos articulaciones giratorias no puede evitar este problema. En este caso en especial, cuando se dirige la ráfaga de balas directamente hacia arriba, en línea recta, su dirección se alinea con el eje de rotación del

acimut. Esto significa que, en este punto en particular, la rotación del acimut no produce un cambio en la dirección del flujo de las balas. Sabemos que necesitamos dos grados de libertad para orientar el flujo de balas pero, en este punto, hemos perdido el uso efectivo de una de las articulaciones. Nuestro mecanismo se ha vuelto **localmente degenerado** en esta ubicación y se comporta como si solamente tuviera un grado de libertad (la dirección de elevación).

Este fenómeno se presenta debido a una **singularidad del mecanismo**. Todos los mecanismos están sujetos a estas dificultades, incluyendo los robots. Al igual que con el mecanismo del artillero de la parte trasera del avión, estas condiciones de singularidad no evitan que el brazo de un robot se posicione en cualquier parte dentro de su espacio de trabajo. Sin embargo, pueden ocasionar problemas con los *movimientos* del brazo en su entorno.

Los manipuladores no siempre se desplazan a través del espacio; algunas veces también se requiere que toquen una pieza o superficie de trabajo y que apliquen una fuerza estática. En este caso surge el siguiente problema: dada una fuerza y un momento de contacto deseados, ¿qué conjunto de **momentos de torsión comunes** se requiere para generarlas? Una vez más, la matriz jacobiana del manipulador surge muy naturalmente para la solución de este problema.

Dinámica

La **dinámica** es un enorme campo dedicado al estudio de las fuerzas que se requieren para ocasionar el movimiento. Para poder acelerar un manipulador desde una posición inerte, deslizarlo a una velocidad constante del efector final y finalmente desacelerarlo hasta detenerlo completamente, los actuadores de las articulaciones deben aplicar un complejo conjunto de funciones de momento de torsión.[2] La forma exacta de las funciones requeridas de momento de torsión de un actuador dependen de los atributos espaciales y temporales de la ruta tomada por el efector final y de las propiedades de masa de los vínculos y de la carga útil; de la fricción en las articulaciones, etcétera. Un método para controlar a un manipulador de manera que siga una ruta deseada implica calcular estas funciones del momento de torsión del actuador, utilizando las ecuaciones dinámicas de movimiento del manipulador.

Muchos de nosotros hemos experimentado la sensación de levantar un objeto que es realmente mucho más ligero de lo que esperábamos (por ejemplo, al levantar un envase de leche del refrigerador, del cual pensábamos que estaba lleno, pero realmente estaba casi vacío). Este mal cálculo de la carga útil puede producir un movimiento de levantamiento inusual. Este tipo de observación indica que el sistema de control humano es más sofisticado que un esquema cinemático puro. Nuestro sistema de control de manipulación hace uso del conocimiento de la masa y otros efectos dinámicos. De igual forma, los algoritmos que construimos para controlar el movimiento del manipulador de un robot deben tomar en cuenta la dinámica.

Un segundo uso de las ecuaciones dinámicas del movimiento es la **simulación**. Al reformular las ecuaciones dinámicas, de manera que la aceleración se calcule como una función del momento de torsión de un actuador, es posible simular cómo se movería un manipulador bajo la aplicación de un conjunto de momentos de torsión de un actuador. (Vea la figura 1.10). A medida que el poder computacional se hace más accesible, crece el uso de las simulaciones y su importancia en muchos campos.

En el capítulo 6 desarrollaremos ecuaciones dinámicas del movimiento, las cuales podrán usarse para controlar o simular el movimiento de los manipuladores.

---

[2] Utilizamos *actuadores de articulaciones* como término genérico para los dispositivos que alimentan a un manipulador; por ejemplo, los motores eléctricos, los actuadores hidráulicos y neumáticos, y los músculos.

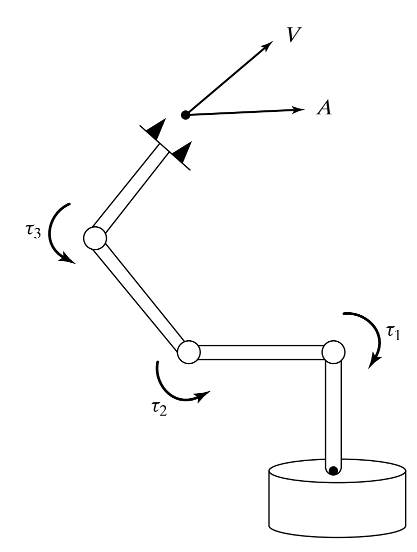


FIGURA 1.10: La relación entre los momentos de torsión aplicados por los actuadores y el movimiento resultante del manipulador se incorpora en las ecuaciones dinámicas de movimiento.

Generación de trayectorias

Una manera común de hacer que un manipulador se mueva de aquí para allá, de una forma suave y controlada, es hacer que cada articulación se mueva según lo especificado por una función continua del tiempo. Comúnmente, cada articulación inicia y termina su movimiento al mismo tiempo, de manera que el movimiento del manipulador parezca coordinado. La manera exacta de calcular estas funciones de movimiento corresponde a la **generación de trayectorias.** (Vea la figura 1.11).

A menudo una ruta se describe no solamente mediante un destino deseado, sino también mediante algunas ubicaciones intermedias, o **puntos vía**, a través de los cuales debe pasar el manipulador en su ruta hacia su destino. En dichas instancias se utiliza algunas veces el término **trazador** para referirse a una función uniforme que pasa por un conjunto de puntos vía.

Para poder forzar al efector final a que siga una línea recta (o cualquier otra figura geométrica) a través del espacio, el movimiento deseado debe convertirse a un conjunto equivalente de movimientos de articulaciones. Esta **generación de trayectorias cartesianas** se considerará también en el capítulo 7.

Diseño y sensores del manipulador

Aunque los manipuladores son, en teoría, dispositivos universales que se aplican a muchas situaciones, la economía generalmente dicta que el dominio de la tarea deseada ejerce una influencia sobre el diseño mecánico del manipulador. Junto a otras cuestiones como el tamaño, la velocidad y la capacidad de carga, el diseñador debe también considerar el número de articulaciones y su arreglo geométrico. Estas consideraciones afectan el tamaño y la calidad del espacio de trabajo del manipulador, la rigidez de la estructura del manipulador y demás atributos.

Entre más articulaciones contenga el brazo de un robot, más destreza y capacidad tendrá. Desde luego que también será más difícil de construir y más costoso. Para

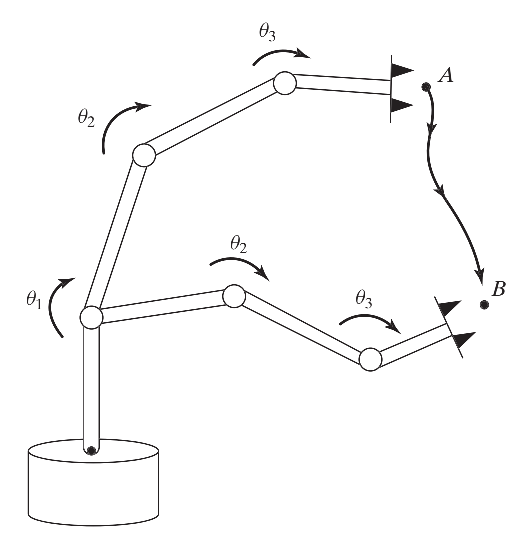


FIGURA 1.11: Para poder mover el efector final a través del espacio, desde el punto A hasta el punto B, debemos calcular una trayectoria para que cada articulación la siga.

poder construir un robot útil, podemos seguir dos enfoques: construir un **robot especializado** para una tarea específica o construir un **robot universal** que pueda realizar una amplia variedad de tareas. En el caso de un robot especializado, debemos pensar cuidadosamente acerca de cuántas articulaciones se necesitan. Por ejemplo, un robot especializado diseñado solamente para colocar componentes electrónicos en un tablero plano de circuitos no necesita tener más de cuatro articulaciones. Tres de ellas permiten que la mano logre obtener cualquier posición en el espacio tridimensional, y la cuarta se agrega para permitir a la mano girar el componente que sujeta sobre un eje vertical. En el caso de un robot universal es interesante observar que las propiedades fundamentales del mundo físico en el que vivimos son las que indican el número mínimo "correcto" de articulaciones; ese número es seis.

Además del diseño del manipulador están las cuestiones relacionadas con la elección y ubicación de los actuadores, los sistemas de transmisión y los sensores de posición interna (y algunas veces de fuerza). (Vea la figura 1.12). Éstas y otras cuestiones de diseño se explicarán en el capítulo 8.

Control de posición lineal

Algunos manipuladores están equipados con motores de pasos o con otros actuadores que pueden ejecutar directamente una trayectoria deseada. Sin embargo, la vasta mayoría de los manipuladores se controlan mediante actuadores que suministran una fuerza o un momento de torsión para ocasionar el movimiento de los vínculos. En este caso se necesita un algoritmo para calcular momentos de torsión que produzcan el movimiento deseado. Este problema de dinámica es fundamental para el diseño de dichos algoritmos, pero no constituye en sí una solución. Algunas de las principales preocupaciones de un **sistema de control de posición** son compensar automáticamente los errores en el conocimiento de los parámetros de un sistema y suprimir los disturbios que tienden a desviar al sistema de la trayectoria deseada. Para lograrlo, el **algoritmo de control** vigila los **sensores** de posición y velocidad, determinando los comandos de momento de

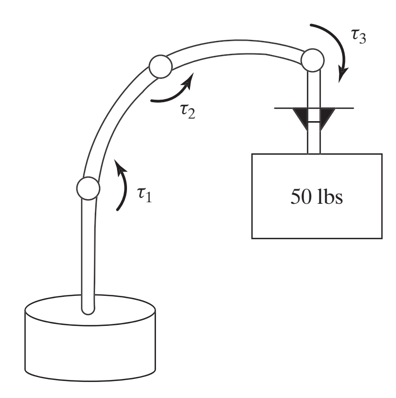


FIGURA 1.12: El diseño de un manipulador mecánico debe considerar la elección del actuador, su ubicación, el sistema de transmisión, la rigidez estructural, la ubicación del sensor y mucho más.

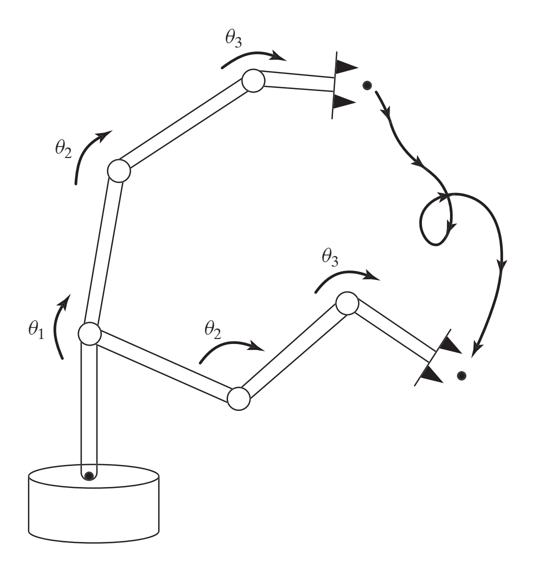


FIGURA 1.13: Para hacer que el manipulador siga la trayectoria deseada, debe implementarse un sistema de control de posición que utilice la retroalimentación de los sensores de las articulaciones para mantener al manipulador en su curso.

torsión para los actuadores. (Vea la figura 1.13). En el capítulo 9 consideraremos los algoritmos de control cuya síntesis se basa en aproximaciones lineales a la dinámica de un manipulador. Estos métodos lineales prevalecen actualmente en la práctica industrial.

Control de posición no lineal

Aunque los sistemas de control basados en los modelos lineales aproximados son populares en los robots industriales de la actualidad, es importante considerar la dinámica no lineal completa del manipulador al sintetizar los algoritmos de control. Actualmente se están creando algunos robots industriales que utilizan algoritmos de **control no lineal** en sus controladores. Estas técnicas no lineales para controlar un

manipulador prometen mejor desempeño que los esquemas lineales más simples. En el capítulo 10 presentaremos los sistemas de control no lineales para los manipuladores mecánicos.

Control de la fuerza

La habilidad de un manipulador para controlar las fuerzas de contacto al tocar piezas, herramientas o superficies de trabajo parece ser de gran importancia al aplicar los manipuladores a muchas tareas del mundo real. El **control de la fuerza** es complementario al control de la posición, ya que generalmente pensamos solamente en uno o en otro para aplicarlo en cierta situación. Cuando se mueve un manipulador en el espacio libre sólo el control de posición tiene sentido, ya que no hay superficie contra la cual reaccionar. Sin embargo, cuando un manipulador está tocando una superficie rígida, los esquemas de control de posición pueden ocasionar la acumulación de fuerzas en el contacto, o pueden hacer que se pierda el contacto con la superficie cuando precisamente eso era lo que se deseaba para alguna aplicación. Los manipuladores raras veces se ven afectados por superficies de reacción en todas las direcciones simultáneamente, por lo que se requiere un control mixto o **híbrido**, en donde algunas direcciones se controlan mediante una **ley de control de posiciones** y las direcciones restantes se controlan mediante una **ley de control de fuerza**. (Vea la figura 1.14). En el capítulo 11 presentaremos una metodología para implementar dicho esquema de control de fuerza.

Si queremos indicarle a un robot que lave una ventana manteniendo cierta fuerza en la dirección perpendicular al plano del vidrio y que siga una trayectoria de movimiento en direcciones tangentes al plano, dichas especificaciones de control dividido o **híbrido** son naturales para estas tareas.

Programación de robots

Un lenguaje de programación de robots sirve como interfaz entre el usuario humano y el robot industrial. Aquí surgen varias preguntas clave: ¿Cómo describe fácilmente el

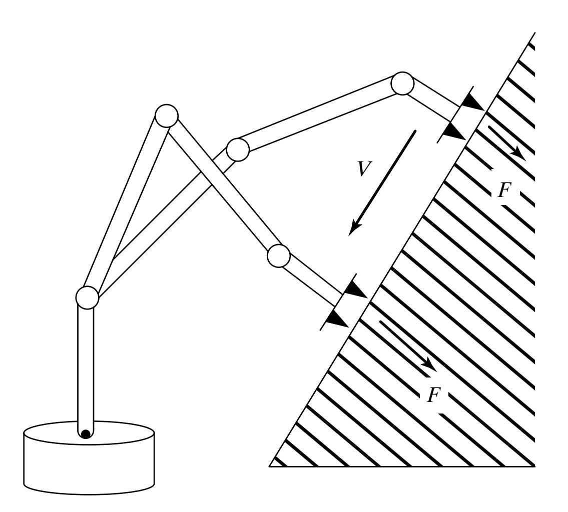


FIGURA 1.14: Para que un manipulador pueda deslizarse a través de una superficie al tiempo que aplica una fuerza constante, debe utilizarse un sistema de control híbrido de posición-fuerza.

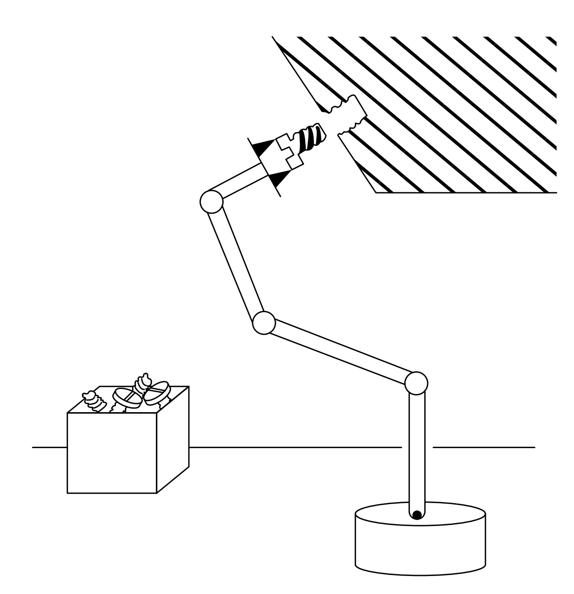


FIGURA 1.15: Los movimientos deseados del manipulador y del efector final, las fuerzas de contacto deseadas y las estrategias de manipulación complejas pueden describirse en un *lenguaje de programación de robots*.

programador los movimientos a través del espacio? ¿Cómo se programan varios manipuladores de manera que puedan funcionar en paralelo? ¿Cómo se describen las acciones basadas en sensores en un lenguaje?

Los manipuladores de robots se diferencian a sí mismos de la **automatización fija** por ser "flexibles", lo que significa que son programables. No sólo son programables los movimientos de los manipuladores sino que, a través del uso de sensores y comunicación con otros tipos de automatización fabril, los manipuladores pueden *adaptarse* a las variaciones a medida que realizan su tarea. (Vea la figura 1.15).

En los sistemas de robots comunes hay una manera rápida de que un usuario humano indique al robot la ruta que debe seguir. En primer lugar, el usuario señala un punto especial en la mano (o tal vez en una herramienta que esté sujetando) como el **punto operacional**, al que algunas veces se le conoce también como **TCP** (punto central de herramienta, por sus siglas en inglés). El usuario debe describir los movimientos del robot en términos de ubicaciones deseadas del punto operacional, relativas a un sistema de coordenadas especificado por el mismo usuario. Generalmente el usuario definirá este sistema de coordenadas de referencia relativo al sistema de coordenadas base del robot, en alguna ubicación relevante para la tarea en cuestión.

Lo más común es que las rutas se construyan mediante la especificación de una secuencia de **puntos vía**. Estos puntos se especifican con relación al sistema de coordenadas de referencia y denotan ubicaciones a lo largo de la ruta a través de la cual debe pasar el TCP. Además de especificar los puntos vía, el usuario puede también indicar que se utilicen ciertas velocidades del TCP en varias porciones de la ruta. Algunas veces pueden también especificarse otros modificadores para modificar el movimiento del robot (por ejemplo, diferentes criterios de uniformidad, etcétera). Con estas "soluciones" el algoritmo de generación de trayectorias debe planear todos los detalles del movimiento: los perfiles de velocidad para las articulaciones, el tiempo de duración

del movimiento, etcétera. Por ende, la solución al problema de generación de trayectorias se proporciona generalmente mediante instrucciones en el lenguaje de programación de robots.

La sofisticación de la interfaz de usuario se está volviendo extremadamente importante a medida que se usan manipuladores y otro tipo de automatización programable en más y más aplicaciones industriales. El problema de programar manipuladores abarca todas las cuestiones de la programación computacional "tradicional" y, por consecuencia, es en sí un tema extenso. Además, ciertos atributos propios del problema de programación de manipuladores hacen que surjan otros contratiempos. Hablaremos sobre algunos de estos temas en el capítulo 12.

Programación fuera de línea y simulación

Un **sistema de programación fuera de línea** es un entorno de programación de robots que se ha extendido lo suficiente, generalmente mediante gráficos computacionales, como para que pueda llevarse a cabo el desarrollo de programas sin la necesidad de acceder al robot en sí. Un argumento común que ha surgido en favor de estos sistemas es que no es necesario detener el equipo de producción (es decir, el robot) para reprogramarlo; por lo tanto, las fábricas automatizadas pueden permanecer en modo de producción una mayor porción del tiempo. (Vea la figura 1.16).

También sirven como un vehículo natural para enlazar las bases de datos de diseño asistido por computadora (CAD) que se utilizan en la fase de diseño de un producto con la fabricación real del mismo. En algunos casos, el uso directo de datos del CAD puede reducir considerablemente el tiempo de programación requerido para el proceso de manufactura. En el capítulo 13 hablaremos sobre los elementos de los sistemas de programación fuera de línea de robots industriales.

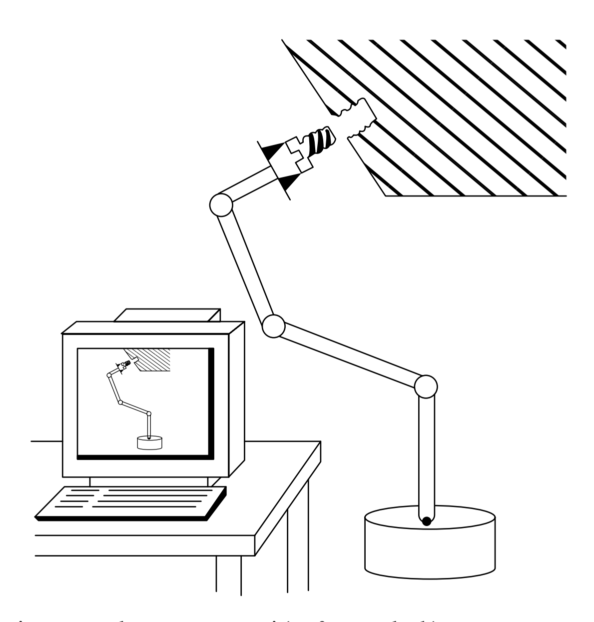


FIGURA 1.16: Los sistemas de programación fuera de línea, que generalmente proporcionan una interfaz de gráficos por computadora, permiten realizar la programación del robot sin necesidad de tener acceso al mismo.

1.3 NOTACIÓN

La notación siempre es una cuestión a tratar en la ciencia y la ingeniería. En este libro utilizaremos las siguientes convenciones:

1. Las variables escritas en mayúsculas generalmente representan vectores o matrices. Las variables en minúsculas representan valores escalares.
2. Los subíndices y superíndices a la izquierda identifican en qué sistema de coordenadas está escrita esa cantidad. Por ejemplo, ${}^{A}P$ representa un vector de posición escrito en el sistema de coordenadas $\{A\}$, y ${}^{A}_{B}R$ es una matriz de rotación[3] que especifica la relación entre los sistemas de coordenadas $\{A\}$ y $\{B\}$.
3. Los superíndices a la derecha se utilizan (según su amplia aceptación) para indicar la inversa o transpuesta de una matriz (por ejemplo, R^{-1}, R^{T}).
4. Los subíndices a la derecha no están sujetos a ninguna convención estricta, pero pueden indicar el componente de un vector (por ejemplo, x, y o z) o pueden usarse como una descripción (como en P_{perno}, la posición de un perno).
5. Utilizaremos muchas funciones trigonométricas. Nuestra notación para el coseno de un ángulo θ_1 puede tomar cualquiera de las siguientes formas: $\cos\theta_1 = c\theta_1 = c_1$.

Los vectores se considerarán como vectores columna; por lo tanto, los vectores fila tendrán la transpuesta indicada explícitamente.

Una observación sobre la notación de vectores en general: muchos textos de mecánica tratan las cantidades vectoriales a un nivel muy abstracto y rutinariamente utilizan vectores definidos en relación con distintos sistemas de coordenadas en las expresiones. El ejemplo más claro es el de la adición de vectores que se dan o se sabe que son relativos a distintos sistemas de referencia. Esto a menudo es muy conveniente ya que produce fórmulas compactas y en parte elegantes. Por ejemplo, considere la velocidad angular, ${}^{0}\omega_4$, del último cuerpo en una conexión en serie de cuatro cuerpos rígidos (como en los vínculos de un manipulador) relativa a la base fija de la cadena. Como las velocidades angulares se suman vectorialmente, podemos escribir una ecuación vectorial muy simple para la velocidad angular del vínculo final:

$$ {}^{0}\omega_4 = {}^{0}\omega_1 + {}^{1}\omega_2 + {}^{2}\omega_3 + {}^{3}\omega_4. \tag{1.1} $$

No obstante y a menos que estas cantidades se expresen respecto a un sistema de coordenadas común, no podrán sumarse; por lo tanto, la ecuación (1.1), aunque elegante, ha ocultado gran parte del “trabajo” del cálculo. Para el caso específico del estudio de los manipuladores mecánicos, ecuaciones como la (1.1) ocultan la tarea de llevar todos los detalles relacionados con los sistemas de coordenadas, que a menudo es con lo que necesitamos tratar en la práctica.

En consecuencia, en este libro llevaremos la información de marco de referencia en la notación para vectores y no debemos sumarlos a menos que se encuentren en el mismo sistema de coordenadas. De esta manera podemos derivar expresiones para resolver el problema de llevar todos los detalles relacionados con los sistemas de coordenadas; además, estas expresiones pueden aplicarse directamente a la computación numérica actual.

BIBLIOGRAFÍA

[1] B. Roth, “Principles of Automation,” *Future Directions in Manufacturing Technology*, basado en el Simposio de la División de Investigación e Ingeniería de Unilever llevado a cabo en Port Sunlight, abril de 1983, publicado por Unilever Research, Reino Unido.

[3] En el capítulo 2 presentaremos este término.

[2] R. Brooks, *Flesh and Machines*, Pantheon Books, Nueva York, 2002.
[3] La Federación Internacional de Robótica y las Naciones Unidas, "Robótica mundial 2001", *Estadísticas, Análisis de mercado, Proyecciones, Casos de Estudio y Rentabilidad de la Inversión en Robots*, Publicación de las Naciones Unidas, Nueva York y Ginebra, 2001.

Libros de referencia general

[4] R. Paul, *Robot Manipulators*, MIT Press, Cambridge, MA, 1981.
[5] M. Brady et al., *Robot Motion*, MIT Press, Cambridge, MA, 1983.
[6] W. Snyder, *Industrial Robots: Computer Interfacing and Control*, Prentice-Hall, Englewood Cliffs, NJ, 1985.
[7] Y. Koren, *Robotics for Engineers*, McGraw-Hill, Nueva York, 1985.
[8] H. Asada y J.J. Slotine, *Robot Analysis and Control*, Wiley, Nueva York, 1986.
[9] K. Fu, R. González y C.S.G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill, Nueva York, 1987.
[10] E. Riven, *Mechanical Design of Robots*, McGraw-Hill, Nueva York, 1988.
[11] J.C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, 1991.
[12] M. Spong, *Robot Control: Dynamics, Motion Planning, and Analysis*, IEEE Press, Nueva York, 1992.
[13] S.Y. Nof, *Handbook of Industrial Robotics*, 2a Edición, Wiley, Nueva York, 1999.
[14] L.W. Tsai, *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*, Wiley, Nueva York, 1999.
[15] L. Sciavicco y B. Siciliano, *Modelling and Control of Robot Manipulators*, 2a Edición, Springer-Verlag, Londres, 2000.
[16] G. Schmierer y R. Schraft, *Service Robots*, A.K. Peters, Natick, Massachussetts, 2000.

Revistas de referencia general

[17] *Robotics World*.
[18] *IEEE Transactions on Robotics and Automation*.
[19] *International Journal of Robotics Research (MIT Press)*.
[20] *ASME Journal of Dynamic Systems, Measurement, and Control*.
[21] *International Journal of Robotics & Automation (IASTED)*.

EJERCICIOS

1.1 [20] Haga una cronología de los principales eventos en el desarrollo de robots industriales durante los últimos 40 años. Consulte la bibliografía y las referencias generales.

1.2 [20] Haga un gráfico que muestre las principales aplicaciones de los robots industriales (por ejemplo: soldadura por puntos, ensamblaje, etcétera) y el porcentaje de robots instalados que se utilizan en cada área de aplicación. Base su gráfico en la información más reciente que pueda encontrar. Consulte la bibliografía y las referencias generales.

1.3 [40] En la figura 1.3 se muestra cómo ha disminuido el costo de los robots industriales con el paso de los años. Busque información sobre el costo de la mano de obra humana en varias industrias específicas (por ejemplo: la mano de obra en la industria automotriz, en la industria de ensamblaje de componentes electrónicos, en la agricultura, etcétera) y haga un gráfico que muestre cómo se comparan estos componentes con el uso de la robótica. En ese gráfico deberá verse que la curva del costo de los robots "cruza" varias de las curvas de costo humano de distintas industrias en distintos momentos. A partir de esta información, derive las fechas aproximadas del momento en que el precio de los robots se hizo rentable para su uso en varias industrias.

1.4 [10] Defina en una o dos oraciones, cinemática, espacio de trabajo y trayectoria.
1.5 [10] Defina en una o dos oraciones estructura, grado de libertad y control de posición.
1.6 [10] Defina en una o dos oraciones control de fuerza y lenguaje de programación de robots.
1.7 [10] Defina en una o dos oraciones control no lineal y programación fuera de línea.
1.8 [20] Haga un gráfico que indique cómo han aumentado los costos de la mano de obra durante los últimos 20 años.
1.9 [20] Genere un gráfico que muestre cómo ha aumentado la razón rendimiento-precio de las computadoras durante los últimos 20 años.
1.10 [20] Genere un gráfico que muestre los principales usos de los robots industriales (por ejemplo: aeroespacial, automotriz, etc.) y el porcentaje de éstos instalados y en uso en cada industria. Base su gráfico en la información más reciente que pueda encontrar.

EJERCICIO DE PROGRAMACIÓN (PARTE 1)

Familiarícese con la computadora que utilizará para realizar los ejercicios de programación al final de cada capítulo. Asegúrese de poder crear y editar archivos y de poder compilar y ejecutar programas.

EJERCICIO MATLAB 1

Al final de la mayoría de los capítulos en este libro viene un ejercicio con MATLAB. En general, estos ejercicios piden al estudiante que programe la matemática de robótica pertinente en MATLAB y que luego revise los resultados de la Caja de Herramientas de Robótica MATLAB (MATLAB Robotics Toolbox). El libro asume que usted está familiarizado con MATLAB y el álgebra lineal (teoría de matrices). Además, el estudiante debe familiarizarse con la Caja de Herramientas de Robótica MATLAB. Para el ejercicio MATLAB 1:

a) De ser necesario, familiarícese con el entorno de programación MATLAB. En la interfaz de comandos del software MATLAB pruebe a escribir *demo* y *help*. Utilizando el editor MATLAB con códigos de colores, aprenda a crear, editar, guardar, ejecutar y depurar archivos-m (archivos ASCII con series de instrucciones MATLAB). Aprenda a crear arreglos (matrices y vectores) y explore las funciones de álgebra lineal de MATLAB para la multiplicación de matrices y vectores, los productos punto y cruz, transpuestas, determinantes e inversas, y para la solución de ecuaciones lineales. MATLAB se basa en el lenguaje C pero, en general, es mucho más fácil de usar. Aprenda a programar instrucciones lógicas y ciclos en MATLAB; a utilizar subprogramas y funciones y a usar comentarios (%) para explicar sus programas y etiquetas para facilitar la legibilidad. Dé un vistazo al sitio www. mathworks.com para obtener más información y tutoriales. Los usuarios avanzados de MATLAB deben familiarizarse con Simulink, la interfaz gráfica de MATLAB, y con la Caja de Herramientas Simbólicas de MATLAB.

b) Familiarícese con la Caja de Herramientas de Robótica MATLAB, un programa con herramientas para MATLAB desarrollado por Peter I. Corke de CSIRO, Pinjarra Hills, Australia, el cual puede descargar sin costo de www.cat.csiro.au/cmst/staff/pic/robot. Puede leer y modificar el código fuente; además, existe una comunidad internacional de usuarios con los que puede comunicarse a través de robot-toolbox@lists.msa.cmst.csiro.au. Descargue la Caja de Herramientas de Robótica MATLAB e instálela en su equipo utilizando el archivo .zip y siguiendo las instrucciones. Lea el archivo *README* y familiarícese con las diversas funciones disponibles para el usuario. Busque el archivo *robot.pdf*; es el manual del usuario que proporciona información sobre las bases y el uso detallado de todas las funciones de este software. No se preocupe si no puede entender el propósito de estas funciones todavía, ya que muchas se relacionan con los conceptos de matemáticas para robots que veremos en los capítulos 2 a 7.

CAPÍTULO 2

Descripciones espaciales y transformaciones

2.1 INTRODUCCIÓN

Por definición, la manipulación robótica implica que se desplazarán piezas y herramientas en el espacio mediante algún tipo de mecanismo. Esto naturalmente conduce a una necesidad de representar posiciones y orientaciones de piezas, herramientas y del mecanismo en sí. Para manipular cantidades mecánicas que representen posición y orientación, debemos definir sistemas de coordenadas y desarrollar convenciones para la representación. Muchas de las ideas aquí desarrolladas en el contexto de posición y orientación formarán una base para nuestra posterior consideración de velocidades lineales y angulares, así como de fuerzas y momentos de torsión.

Adoptaremos la filosofía de que en alguna parte existe un **sistema de coordenadas universal**, y que todo lo que hablemos puede hacer referencia a este sistema. Describiremos todas las posiciones y orientaciones respecto al sistema de coordenadas universal o respecto a otros sistemas de coordenadas cartesianas que se definen (o podrían definirse) en forma relativa al sistema universal.

2.2 DESCRIPCIONES: POSICIONES, ORIENTACIONES Y TRAMAS

Una **descripción** se utiliza para especificar los atributos de varios objetos con los que trata un sistema de manipulación. Estos objetos son piezas, herramientas y el manipulador en sí. En esta sección hablaremos sobre la descripción de posiciones, orientaciones y de una entidad que contiene ambas descripciones: la trama.

Descripción de una posición

Una vez que se establece un sistema de coordenadas, podemos ubicar cualquier punto en el universo con un **vector de posición** de orden 3×1. Como es común que definamos muchos sistemas de coordenadas, además del sistema de coordenadas universal, los vectores deben etiquetarse con información que identifique en cuál sistema están definidos. En este libro los vectores se escriben con un subíndice a la izquierda que indica el sistema de coordenadas al que hacen referencia (a menos que quede claro, según el contexto); por ejemplo, ${}^{A}P$. Esto significa que los componentes de ${}^{A}P$ tienen valores numéricos que indican distancias sobre los ejes de $\{A\}$. Cada una de estas distancias sobre un eje puede definirse como el resultado de proyectar el vector sobre el eje correspondiente.

La figura 2.1 representa el dibujo de un sistema de coordenadas llamado $\{A\}$, con tres vectores unitarios mutuamente ortogonales con puntas sólidas. Un punto ${}^{A}P$ se representa como un vector y puede definirse de manera equivalente como una posición en el espacio, o simplemente como un conjunto ordenado de tres números. Dados los subíndices x, y y z, los elementos individuales de un vector son:

$$ {}^{A}P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{2.1} $$

En resumen, describiremos la posición de un punto en el espacio con un vector de posición. Otras descripciones de la posición de puntos en tres dimensiones, como las representaciones en coordenadas esféricas o cilíndricas, se discutirán en los ejercicios al final del capítulo.

Descripción de una orientación

A menudo necesitamos representar no solamente un punto en el espacio, sino también describir la **orientación** de un cuerpo en el espacio. Por ejemplo, si el vector ${}^{A}P$ de la figura 2.2 ubica el punto directamente entre las puntas de los dedos de la mano de un manipulador, la ubicación completa de la mano no se especifica sino hasta que se proporciona también su orientación. Suponiendo que el manipulador tiene un número suficiente de articulaciones,[1] la mano podría *orientarse* arbitrariamente y al mismo tiempo podría mantenerse el punto entre las puntas de los dedos en la misma posición

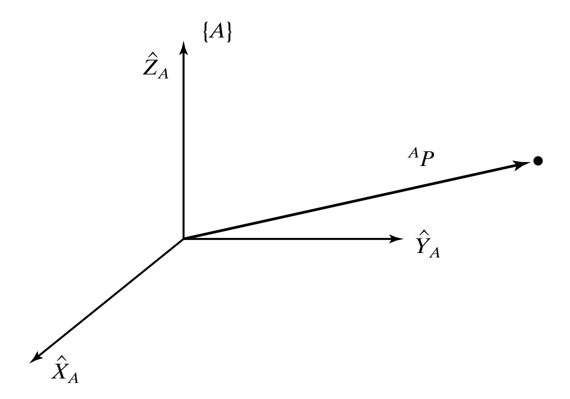


FIGURA 2.1: Vector relativo a la trama (ejemplo).

[1] En los capítulos 3 y 4 hablaremos sobre cuántas articulaciones son “suficientes”.

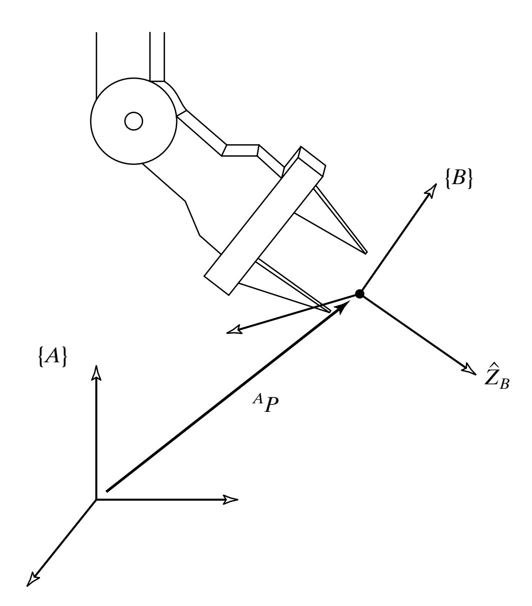


FIGURA 2.2: Ubicación de un objeto en posición y orientación.

en el espacio. Para describir la orientación de un cuerpo, *adjuntaremos un sistema de coordenadas al cuerpo y luego daremos una descripción de este sistema de coordenadas relativo al sistema de referencia*. En la figura 2.2 se ha adjuntado el sistema de coordenadas $\{B\}$ al cuerpo de una manera conocida. Ahora basta con una descripción de $\{B\}$ relativo a $\{A\}$ para dar la orientación del cuerpo.

Por ende, las posiciones de los puntos se describen con vectores, y las orientaciones de los cuerpos con un sistema de coordenadas adjunto. Una manera de describir el sistema de coordenadas $\{B\}$ adjunto al cuerpo es escribiendo los vectores unitarios de sus tres ejes principales[2] en términos del sistema de coordenadas $\{A\}$.

Para denotar los vectores unitarios proporcionamos las direcciones principales del sistema de coordenadas $\{B\}$ como $\hat{X}_B$, $\hat{Y}_B$ y $\hat{Z}_B$. Al escribirse en términos del sistema de coordenadas $\{A\}$, se llaman $^A\hat{X}_B$, $^A\hat{Y}_B$ y $^A\hat{Z}_B$. Es conveniente si apilamos estos tres vectores unitarios como columnas de una matriz de 3×3, en el orden $^A\hat{X}_B$, $^A\hat{Y}_B$, $^A\hat{Z}_B$. A esta matriz la llamaremos **matriz de rotación** y, dado que esta matriz de rotación específica describe a $\{B\}$ en forma relativa a $\{A\}$, la representamos con la notación ^A_BR (la elección de los superíndices y subíndices a la izquierda en la definición de las matrices de rotación quedará clara en las siguientes secciones):

$$^A_BR = \left[\ ^A\hat{X}_B \quad ^A\hat{Y}_B \quad ^A\hat{Z}_B\ \right] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \tag{2.2}$$

En resumen, puede utilizarse un conjunto de tres vectores para especificar una orientación. A fin de ser más claros, construiremos una matriz de 3×3 que tenga estos tres vectores como columnas. En consecuencia, siempre que se represente la posición de un punto con un vector, la orientación de un cuerpo se representará con una matriz. En la

[2] A menudo es conveniente utilizar tres, aunque con dos bastaría. (El tercero puede recuperarse tomando el producto cruz de los dos ejes dados).

sección 2.8 consideraremos otras descripciones de orientación que requieren solamente tres parámetros.

Podemos ofrecer expresiones para los escalares r_{ij} en (2.2) teniendo en cuenta que los componentes de cualquier vector son simplemente las proyecciones de ese vector en las direcciones unitarias de su trama de referencia. Por lo tanto, cada componente de ${}^A_B R$ en (2.2) puede escribirse como el producto punto de un par de vectores unitarios:

$$ {}^A_B R = \left[\, {}^A\hat{X}_B \quad {}^A\hat{Y}_B \quad {}^A\hat{Z}_B \, \right] = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix}. \tag{2.3} $$

Por cuestión de brevedad hemos omitido los superíndices a la izquierda de la matriz más a la derecha de (2.3). De hecho, la elección de la trama en la cual se van a describir los vectores unitarios es arbitraria, siempre y cuando sea la misma para cada par que se obtenga del producto punto. Como el producto punto de dos vectores unitarios es el coseno del ángulo entre ellos, queda claro por qué a los componentes de las matrices de rotación se les llama comúnmente **cosenos de dirección**.

Una inspección más detallada de (2.3) nos muestra que las filas de la matriz son los vectores unitarios de $\{A\}$ expresados en $\{B\}$; es decir,

$$ {}^A_B R = \left[\, {}^A\hat{X}_B \quad {}^A\hat{Y}_B \quad {}^A\hat{Z}_B \, \right] = \begin{bmatrix} {}^B\hat{X}_A^T \\ {}^B\hat{Y}_A^T \\ {}^B\hat{Z}_A^T \end{bmatrix}. \tag{2.4} $$

Así, ${}^A_B R$, la descripción de la trama $\{A\}$ relativa a $\{B\}$, se da mediante la transpuesta de (2.3); esto es,

$$ {}^B_A R = {}^A_B R^T. \tag{2.5} $$

Esto sugiere que la inversa de una matriz de rotación es igual a su transpuesta; lo cual puede verificarse fácilmente de la siguiente manera

$$ {}^A_B R^T \, {}^A_B R = \begin{bmatrix} {}^A\hat{X}_B^T \\ {}^A\hat{Y}_B^T \\ {}^A\hat{Z}_B^T \end{bmatrix} \left[\, {}^A\hat{X}_B \quad {}^A\hat{Y}_B \quad {}^A\hat{Z}_B \, \right] = I_3, \tag{2.6} $$

Donde I_3 es la matriz identidad 3×3. Por lo que,

$$ {}^A_B R = {}^B_A R^{-1} = {}^B_A R^T. \tag{2.7} $$

En efecto, mediante el álgebra lineal [1], sabemos que el inverso de una matriz con columnas ortonormales es igual a su transpuesta. Acabamos de demostrar esto geométricamente.

Descripción de una trama

La información necesaria para especificar completamente en dónde se encuentra la mano del manipulador en la figura 2.2 es una posición y una orientación. El punto en el cuerpo cuya posición describimos podría elegirse arbitrariamente. *Por conveniencia, el punto cuya posición describiremos se elige como el origen de la trama adjunta al cuerpo*.

La situación de un par posición y orientación surge tan a menudo en robótica que definimos una entidad llamada **trama**, la cual es un conjunto de cuatro vectores que proporcionan información sobre la posición y la orientación. Por ejemplo, en la figura 2.2 un vector ubica la posición de la punta de los dedos y tres más describen su orientación. De manera equivalente, la descripción de una trama puede definirse como un vector de posición y una matriz de rotación. Observe que una trama es un sistema de coordenadas en donde además de la orientación damos un vector de posición que ubica su origen de manera relativa a alguna otra trama fija. Por ejemplo, la trama $\{B\}$ se describe mediante ${}^{A}_{B}R$ y ${}^{A}P_{BORG}$, en donde ${}^{A}P_{BORG}$ es el vector que ubica el origen de la trama $\{B\}$:

$$\{B\} = \{{}^{A}_{B}R, {}^{A}P_{BORG}\}. \tag{2.8}$$

En la figura 2.3 hay tres tramas que se muestran junto con el sistema de coordenadas universal. Se sabe que las tramas $\{A\}$ y $\{B\}$ son relativas al sistema de coordenadas universal y que la trama $\{C\}$ se conoce en relación a la trama $\{A\}$.

En la figura 2.3 presentamos una *representación gráfica* de las tramas, lo cual es conveniente para visualizarlas. Una trama se describe mediante tres flechas que representan vectores unitarios, los cuales definen los ejes principales de la trama. Es necesario dibujar una flecha que representa a un vector que va de un origen hacia el otro. Este vector representa la posición del origen en la punta de la flecha, en términos de la trama en la cola de la flecha. Por ejemplo, en la figura 2.3, la dirección de esta flecha de ubicación nos dice que se sabe que $\{C\}$ se conoce en términos de $\{A\}$ y no al revés.

En resumen, una trama puede utilizarse como una descripción de un sistema de coordenadas relativo a otro. Una trama abarca dos ideas al representar tanto la posición como la orientación, por lo que puede definirse como una generalización de esas dos ideas. Las posiciones podrían representarse mediante una trama cuya parte correspondiente a la matriz de rotación es la matriz identidad y cuya parte correspondiente al vector de posición ubica el punto que se está describiendo. De igual forma, una orientación podría representarse mediante una trama cuya parte correspondiente al vector de posición sea al vector cero.

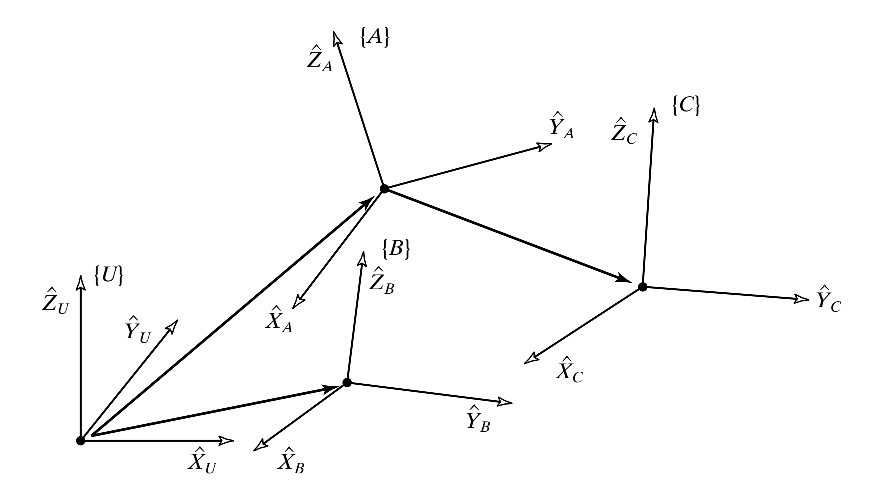


FIGURA 2.3: Ejemplo de varias tramas.

2.3 ASIGNACIONES: CÓMO CAMBIAR DESCRIPCIONES DE TRAMA A TRAMA

Gran parte de los problemas de la robótica se refieren a cómo expresar la misma cantidad en varios sistemas de coordenadas de referencia. En la sección anterior presentamos descripciones de posiciones, orientaciones y tramas; ahora consideraremos las matemáticas de la **asignación** o **mapeo**, para poder cambiar las descripciones de una trama a otra.

Asignaciones que involucran tramas trasladadas

En la figura 2.4 tenemos una posición definida por el vector ${}^{B}P$. Deseamos expresar este punto en el espacio en términos de la trama $\{A\}$, cuando $\{A\}$ tenga la misma orientación que $\{B\}$. En este caso $\{B\}$ difiere de $\{A\}$ sólo por una *traslación*, que se da mediante ${}^{A}P_{BORG}$, un vector que ubica el origen de $\{B\}$ relativo a $\{A\}$.

Como ambos vectores se definen en relación a las tramas de la misma orientación, calculamos la descripción del punto P relativo a $\{A\}$, ${}^{A}P$, mediante la suma de vectores:

$$ {}^{A}P = {}^{B}P + {}^{A}P_{BORG}. \tag{2.9} $$

Observe que sólo en el caso especial de orientaciones equivalentes podemos sumar vectores que estén definidos en términos de tramas diferentes.

En este ejemplo simple, hemos ilustrado la **asignación** de un vector de una trama a otra. Esta idea de asignar, o cambiar la descripción de una trama a otra, es un concepto extremadamente importante. La cantidad en sí (en este caso, un punto en el espacio) no se modifica; sólo se modifica su descripción. Esto se ilustra en la figura 2.4, en donde el punto descrito por ${}^{B}P$ no se traslada sino que permanece igual, y lo que se ha hecho es calcular una nueva descripción del mismo punto con respecto al sistema $\{A\}$.

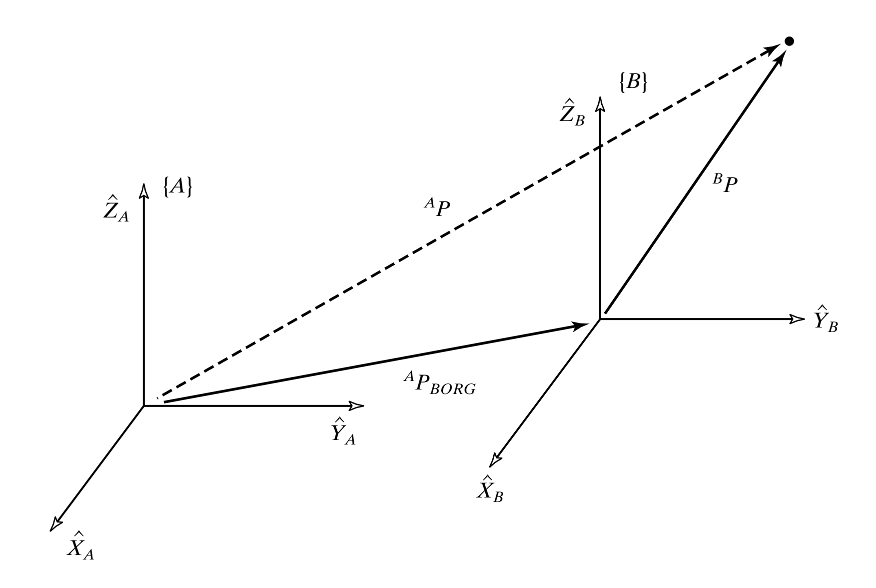


FIGURA 2.4: Asignación de traslación.

Decimos que el vector ${}^{A}P_{BORG}$ define esta asignación porque toda la información necesaria para realizar la modificación en la descripción está contenida en ${}^{A}P_{BORG}$ (junto con el conocimiento de que las tramas tenían una orientación equivalente).

Asignaciones que involucran tramas rotadas

En la sección 2.2 introdujimos la noción de describir una orientación mediante tres vectores unitarios, denotando los ejes principales de un sistema de coordenadas adjunto a un cuerpo. Por conveniencia apilamos estos tres vectores unitarios como columnas de una matriz 3×3. A esta matriz la llamaremos matriz de rotación y, si esta matriz de rotación específica describe a $\{B\}$ en términos de $\{A\}$, la identificamos con la notación ${}^{A}_{B}R$.

Observe que, según nuestra definición, todas las columnas de una matriz de rotación tienen una magnitud unitaria, además de que estos vectores unitarios son ortogonales. Como vimos antes, una consecuencia de esto es que:

$$ {}^{A}_{B}R = {}^{B}_{A}R^{-1} = {}^{B}_{A}R^{T}. \tag{2.10} $$

Por lo tanto, y como las columnas de ${}^{A}_{B}R$ son los vectores unitarios de $\{B\}$ escritos en $\{A\}$, las *filas* de ${}^{A}_{B}R$ son los vectores unitarios de $\{A\}$ escritos en $\{B\}$.

De esta manera, una matriz de rotación puede interpretarse como un conjunto de tres vectores columna o como un conjunto de tres vectores fila, de la siguiente manera:

$$ {}^{A}_{B}R = \left[\ {}^{A}\hat{X}_{B} \quad {}^{A}\hat{Y}_{B} \quad {}^{A}\hat{Z}_{B}\ \right] = \begin{bmatrix} {}^{B}\hat{X}_{A}^{T} \\ {}^{B}\hat{Y}_{A}^{T} \\ {}^{B}\hat{Z}_{A}^{T} \end{bmatrix}. \tag{2.11} $$

Como en la figura 2.5, es común que se presente la situación en la que conocemos la definición de un vector con respecto a cierta trama, $\{B\}$, y nos gustaría conocer su definición con respecto a otra trama, $\{A\}$, en donde los orígenes de las dos tramas sean

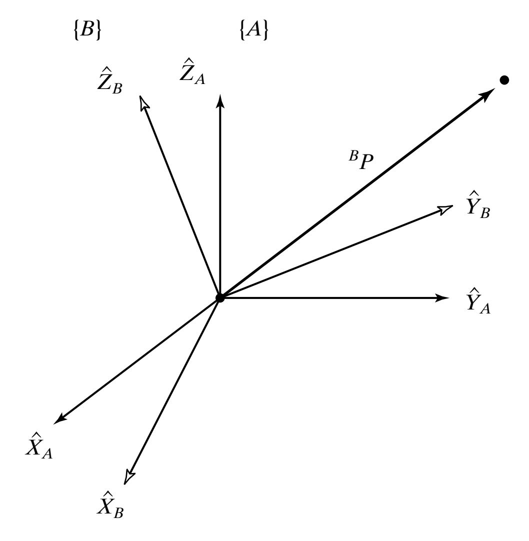


FIGURA 2.5: Rotación de la descripción de un vector.

coincidentes. Este cálculo es posible cuando se conoce una descripción de la orientación de $\{B\}$ relativa a $\{A\}$. Esta orientación se representa mediante la matriz de rotación ${}^{A}_{B}R$, cuyas columnas son los vectores unitarios de $\{B\}$ escritos en $\{A\}$.

Para poder calcular ${}^{A}P$, hay que tener en cuenta que los componentes de cualquier vector son simplemente las proyecciones de ese vector en las direcciones unitarias de su trama. La proyección se calcula como el producto punto de los vectores. De esta manera, podemos ver que los componentes de ${}^{A}P$ pueden calcularse como:

$$\begin{aligned} {}^{A}px &= {}^{B}\hat{X}_{A} \cdot {}^{B}P, \\ {}^{A}p_{y} &= {}^{B}\hat{Y}_{A} \cdot {}^{B}P, \\ {}^{A}p_{z} &= {}^{B}\hat{Z}_{A} \cdot {}^{B}P. \end{aligned} \tag{2.12}$$

Para poder expresar la ecuación (2.12) en términos de una multiplicación de matrices de rotación, hay que observar en la ecuación (2.11) que las *filas* de ${}^{A}_{B}R$ son ${}^{B}X_{A}$, ${}^{B}Y_{A}$ y ${}^{B}Z_{A}$. Así que la ecuación (2.12) puede escribirse en forma compacta, utilizando una matriz de rotación:

$$ {}^{A}P = {}^{A}_{B}R \; {}^{B}P. \tag{2.13}$$

La ecuación (2.13) implementa una asignación; es decir, modifica la descripción de un vector: de ${}^{B}P$, que describe un punto en el espacio relativo a $\{B\}$, a ${}^{A}P$, que es una descripción del mismo punto, pero expresado en forma relativa a $\{A\}$.

Ahora podemos ver que nuestra notación es de gran ayuda para llevar el registro de las asignaciones y las tramas de referencia. Una manera útil de visualizar la notación que hemos introducido es imaginar que los subíndices a la izquierda cancelan los superíndices a la izquierda de la siguiente entidad, por ejemplo las B en (2.13).

EJEMPLO 2.1

La figura 2.6 muestra una trama $\{B\}$ que se gira 30 grados de manera relativa a la trama $\{A\}$ sobre $\hat{Z}$. Aquí, $\hat{Z}$ apunta hacia fuera de la página.

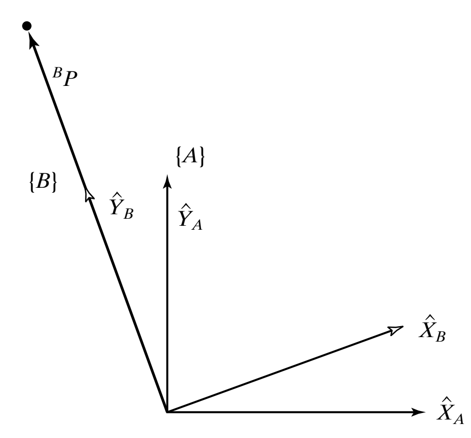


FIGURA 2.6: $\{B\}$ con un giro de 30 grados sobre $\hat{Z}$.

Si escribimos los vectores unitarios de $\{B\}$ en términos de $\{A\}$ y los apilamos como columnas de la matriz de rotación, obtendremos

$$ {}^{A}_{B}R = \begin{bmatrix} 0.866 & -0.500 & 0.000 \\ 0.500 & 0.866 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}. \tag{2.14} $$

Dado que

$$ {}^{B}P = \begin{bmatrix} 0.0 \\ 2.0 \\ 0.0 \end{bmatrix}, \tag{2.15} $$

calculamos ${}^{A}P$ como

$$ {}^{A}P = {}^{A}_{B}R\,{}^{B}P = \begin{bmatrix} -1.000 \\ 1.732 \\ 0.000 \end{bmatrix}. \tag{2.16} $$

Aquí, ${}^{A}_{B}R$ actúa como una asignación que se utiliza para describir a ${}^{B}P$ en forma relativa a la trama $\{A\}$, ${}^{A}P$. Como en el caso de las traslaciones, es importante recordar que, si se ve como una asignación, el vector original P no cambia en el espacio. En vez de ello, calculamos una nueva descripción del vector en relación a otra trama.

Asignaciones que involucran tramas arbitrarias

Muy a menudo, nos encontramos con una situación en la que conocemos la descripción de un vector respecto a cierta trama $\{B\}$, y es conveniente conocer su descripción con respecto a otra trama, $\{A\}$. Ahora consideraremos el caso general de la asignación. Aquí el origen de la trama $\{B\}$ no es coincidente con el de la trama $\{A\}$, sino que tiene un desplazamiento vectorial arbitrario. El vector que define el origen de $\{B\}$ se llama ${}^{A}P_{BORG}$. Además, $\{B\}$ se gira con respecto a $\{A\}$, según lo describe ${}^{A}_{B}R$. Dado ${}^{B}P$, deseamos calcular ${}^{A}P$, como en la figura 2.7.

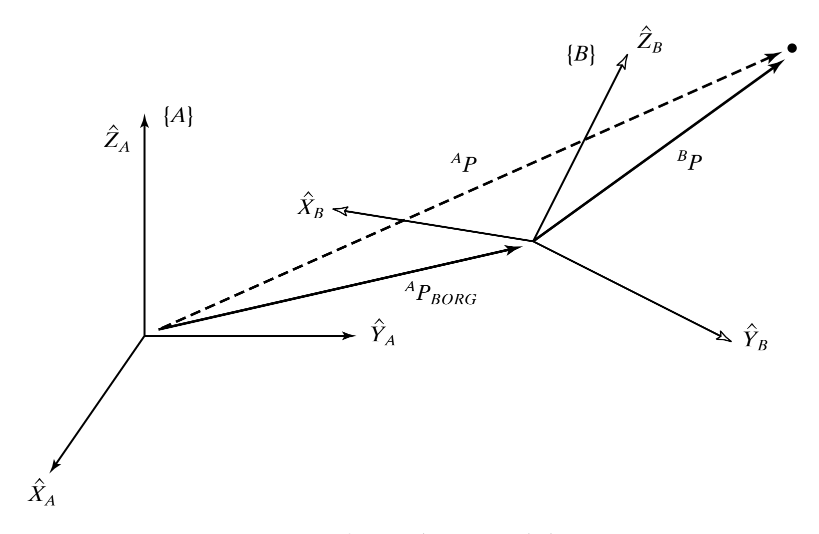


FIGURA 2.7: Transformada general de un vector.

Primero podemos cambiar ${}^{B}P$ a su descripción relativa a una trama intermedia que tiene la misma orientación que $\{A\}$, pero cuyo origen es coincidente con el origen de $\{B\}$. Esto se hace multiplicándolo previamente por ${}^{A}_{B}R$, como en la última sección. Después consideramos la traslación entre orígenes mediante una simple suma vectorial, como antes, y obtenemos:

$$ {}^{A}P = {}^{A}_{B}R\,{}^{B}P + {}^{A}P_{BORG}. \tag{2.17} $$

La ecuación (2.17) describe una asignación de transformación general de un vector a partir de su descripción en una trama a una descripción en una segunda trama. Observe la siguiente interpretación de nuestra notación según se muestra en (2.17): las "B" se cancelan, dejando todas las cantidades como vectores escritos en términos de "A", los cuales pueden ya sumarse sin problema.

La forma de (2.17) no es tan atractiva como la forma conceptual

$$ {}^{A}P = {}^{A}_{B}T\,{}^{B}P. \tag{2.18} $$

Esto es, nos gustaría pensar en una asignación de una trama a otra como un operador en forma de matriz. Esto nos ayuda a escribir ecuaciones compactas y es conceptualmente más claro que (2.17). Para poder escribir las ecuaciones matemáticas dadas en (2.17) en la forma de operador matricial sugerida por (2.18), definimos un operador matricial de 4×4 y utilizamos vectores de posición de 4×1, para que (2.18) tenga la siguiente estructura:

$$ \begin{bmatrix} {}^{A}P \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} & {}^{A}_{B}R & & {}^{A}P_{BORG} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} {}^{B}P \\ 1 \end{bmatrix}. \tag{2.19} $$

En otras palabras:

1. Se agrega un "1" como el último elemento de los vectores de 4×1.
2. Se agrega una fila "[0 0 0 1]" como la última fila de la matriz de 4×4.

Adoptaremos la convención de que un vector de posición es de 3×1 o 4×1, dependiendo de si se multiplica por una matriz de 3×3 o por una de 4×4. Puede verse de antemano que (2.19) implementa lo siguiente:

$$ \begin{aligned} {}^{A}P &= {}^{A}_{B}R\,{}^{B}P + {}^{A}P_{BORG} \\ 1 &= 1. \end{aligned} \tag{2.20} $$

La matriz de 4×4 en (2.19) se llama **transformada homogénea**. Para nuestros fines, puede tratarse solamente como una construcción utilizada para convertir la rotación y la traslación de la transformada general en una sola forma matricial. En otros campos de estudio puede utilizarse para calcular operaciones de perspectiva y escalado (cuando la última fila es distinta de "[0 0 0 1]" o la matriz de rotación no es ortonormal). El lector interesado debería consultar [2].

Comúnmente escribiremos una ecuación como (2.18) sin ninguna notación que indique que es una representación homogénea, ya que esto es obvio debido al contexto. Observe que aunque las transformadas homogéneas son útiles para escribir ecuaciones compactas, un programa computacional para transformar vectores generalmente no las utilizaría debido al tiempo que se pierde al multiplicar unos y ceros. Por ende, esta representación es principalmente para nuestra conveniencia al idear y escribir ecuaciones en papel.

Así como utilizamos matrices de rotación para especificar una orientación, utilizaremos transformadas (usualmente en representación homogénea) para especificar una trama. Observe que aunque hemos introducido transformadas homogéneas en el contexto de las asignaciones, también sirven como descripciones de tramas. La descripción de la trama $\{B\}$ relativa a $\{A\}$ es ${}^A_BT$.

EJEMPLO 2.2

La figura 2.8 muestra una trama $\{B\}$ que se gira 30 grados en forma relativa a $\{A\}$ sobre $\hat{Z}$, se trasladada 10 unidades en $\hat{X}_A$ y 5 unidades en $\hat{Y}_A$. Encuentre ${}^AP$ en donde ${}^BP = [3.0\ 7.0\ 0.0]^T$.

La definición de la trama $\{B\}$ es

$$ {}^A_BT = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 10.0 \\ 0.500 & 0.866 & 0.000 & 5.0 \\ 0.000 & 0.000 & 1.000 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.21} $$

Dada

$$ {}^BP = \begin{bmatrix} 3.0 \\ 7.0 \\ 0.0 \end{bmatrix}, \tag{2.22} $$

utilizamos la definición de $\{B\}$ que se acaba de dar como una transformación:

$$ {}^AP = {}^A_BT\ {}^BP = \begin{bmatrix} 9.098 \\ 12.562 \\ 0.000 \end{bmatrix}. \tag{2.23} $$

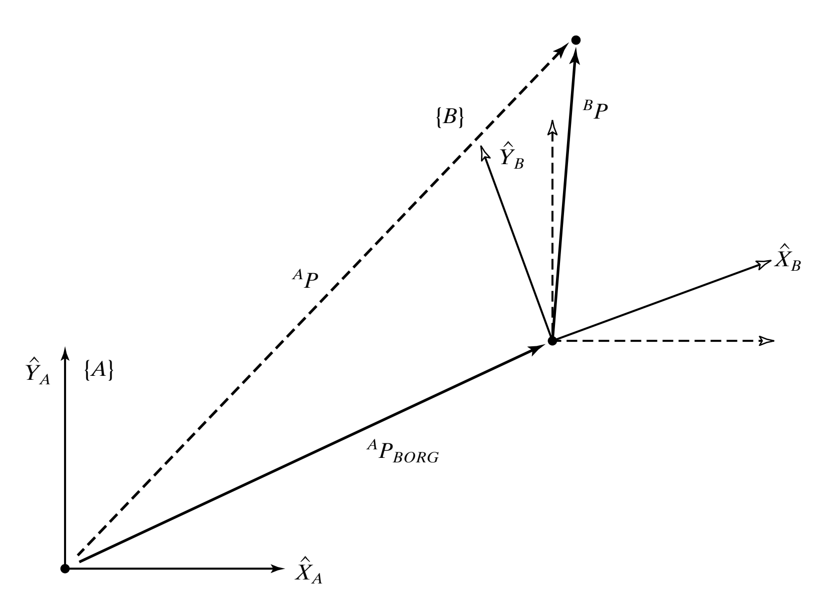


FIGURA 2.8: La trama $\{B\}$ girada y trasladada.

2.4 OPERADORES: TRASLACIONES, ROTACIONES Y TRANSFORMACIONES

Las mismas formas matemáticas que se utilizan para correlacionar puntos entre tramas pueden interpretarse también como operadores que trasladan puntos, giran vectores o hacen ambas cosas. En esta sección ilustraremos esta interpretación de las matemáticas que ya hemos desarrollado.

Operadores de traslación

Una traslación desliza un punto en el espacio una distancia finita a lo largo de una dirección vectorial dada. Con esta interpretación de trasladar el punto en el espacio, sólo necesita estar involucrado un sistema de coordenadas. Resulta que el proceso de trasladar el punto en el espacio se logra con las mismas matemáticas utilizadas para asignar el punto a una segunda trama. Casi siempre es muy importante comprender cuál interpretación de las matemáticas se está utilizando. La distinción es tan simple como esto: cuando un vector se desplaza "hacia adelante" en forma relativa a una trama, podemos considerar bien que el vector se desplazó "hacia adelante" o que la trama se movió "hacia atrás". Las matemáticas involucradas en ambos casos son idénticas; sólo la manera en que vemos la situación es distinta. La figura 2.9 indica gráficamente cómo se traslada un vector ${}^{A}P_{1}$ mediante un vector ${}^{A}Q$. Aquí el vector ${}^{A}Q$ proporciona la información necesaria para realizar la traslación.

El resultado de la operación es un nuevo vector ${}^{A}P_{2}$ que se calcula así:

$${}^{A}P_{2} = {}^{A}P_{1} + {}^{A}Q. \tag{2.24}$$

Para escribir esta operación de traslación como un operador matricial utilizamos la notación:

$${}^{A}P_{2} = D_{Q}(q)\,{}^{A}P_{1}, \tag{2.25}$$

en donde q es la magnitud con signo de la traslación a lo largo de la dirección vectorial $\hat{Q}$. El operador D_{Q} puede considerarse como una transformada homogénea de una

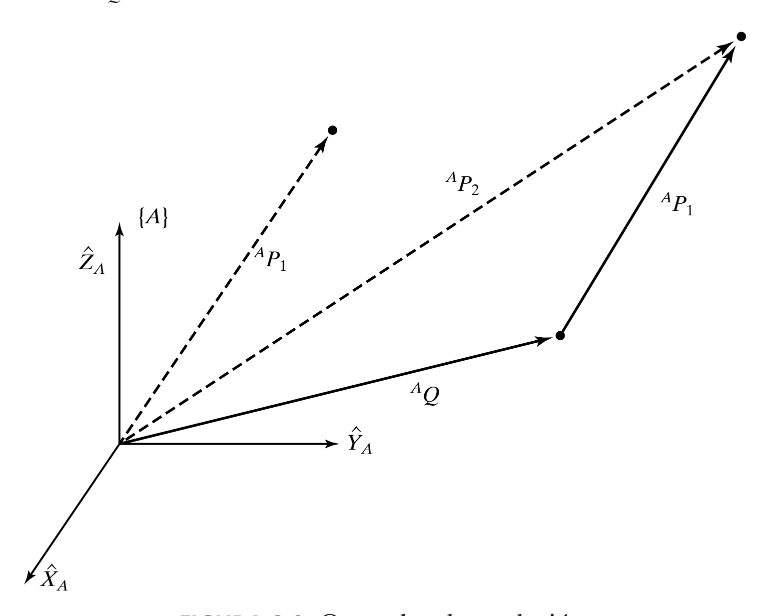


FIGURA 2.9: Operador de traslación.

forma especialmente simple:

$$D_Q(q) = \begin{bmatrix} 1 & 0 & 0 & q_x \\ 0 & 1 & 0 & q_y \\ 0 & 0 & 1 & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.26}$$

en donde q_x, q_y y q_z son los componentes del vector de traslación Q y $q = \sqrt{q_x^2 + q_y^2 + q_z^2}$. Las ecuaciones (2.9) y (2.24) implementan las mismas matemáticas. Observe que si hubiéramos definido a ${}^BP_{AORG}$ (en vez de ${}^AP_{BORG}$) en la figura 2.4 y lo hubiéramos usado en (2.9), habríamos visto un cambio de signo entre (2.9) y (2.24). Este cambio de signo indicaría la diferencia entre desplazar el vector "hacia adelante" y desplazar el sistema de coordenadas "hacia atrás". Al definir la ubicación de $\{B\}$ en forma relativa a $\{A\}$ (con ${}^AP_{BORG}$), hacemos que las matemáticas de las dos interpretaciones sean idénticas. Ahora que hemos introducido la notación "D_Q", podemos también usarla para describir tramas y como una asignación.

Operadores rotacionales

También se puede interpretar a una matriz de rotación como un operador rotacional que opera sobre un vector ${}^AP_1$ y convierte ese vector en uno nuevo, ${}^AP_2$, por medio de una rotación R. Generalmente, cuando una matriz de rotación se muestra como un operador, no aparecen subíndices ni superíndices, ya que no se considera que esté relacionando dos tramas. Esto es, podemos escribir

$${}^AP_2 = R\,{}^AP_1. \tag{2.27}$$

De nuevo y como en el caso de las traslaciones, los cálculos descritas en (2.13) y en (2.27) son las mismas; sólo nuestra interpretación es distinta. Este hecho también nos permite ver *cómo obtener* matrices rotacionales que vayan a utilizarse como operadores:

La matriz de rotación que gira vectores a través de cierta rotación R, es la misma que la matriz de rotación que describe a una trama girada por R en relación con la trama de referencia.

Aunque una matriz de rotación puede verse fácilmente como un operador, también definiremos otra notación para un operador rotacional que indica claramente sobre qué eje se está girando:

$${}^AP_2 = R_K(\theta)\,{}^AP_1. \tag{2.28}$$

En esta notación, "$R_K(\theta)$" es un operador rotacional que realiza una rotación de θ grados sobre la dirección del eje $\hat{K}$. Este operador puede escribirse como una transformada homogénea cuya parte correspondiente al vector de posición sea igual a cero. Por ejemplo, si sustituimos en (2.11) se produce el operador que gira θ grados sobre el eje $\hat{Z}$:

$$R_z(\Theta) = \begin{bmatrix} \cos\theta & -\operatorname{sen}\theta & 0 & 0 \\ \operatorname{sen}\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.29}$$

Claro que, para girar un vector de posición, podríamos utilizar también la parte correspondiente a la matriz de rotación de orden 3×3 de la transformada homogénea. Por lo tanto, se puede considerar que la notación "R_K" representa a una matriz de 3×3 o de 4×4. Más adelante veremos cómo escribir la matriz de rotación para una rotación sobre un eje $\hat{K}$ arbitrario.

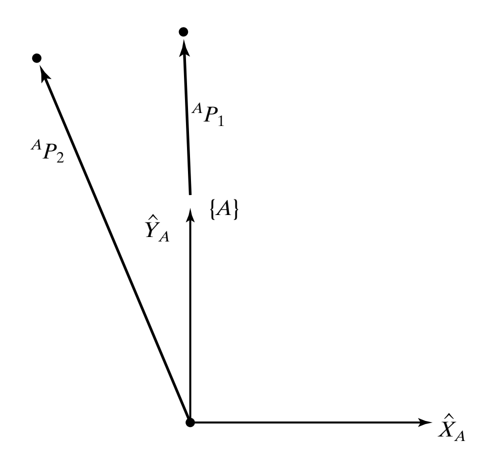


FIGURA 2.10: El vector $^AP_1$ girado 30 grados sobre $\hat{Z}$.

EJEMPLO 2.3

La figura 2.10 muestra un vector $^AP_1$. Deseamos calcular el vector que se obtiene al girar este vector 30 grados sobre $\hat{Z}$. Llamaremos a este nuevo vector $^AP_2$.

La matriz de rotación que gira los vectores 30 grados sobre $\hat{Z}$ es la misma que la matriz de rotación que describe a una trama que se gira 30 grados sobre $\hat{Z}$ con respecto a la trama de referencia. Por lo tanto, el operador rotacional correcto es

$$R_z(30.0) = \begin{bmatrix} 0.866 & -0.500 & 0.000 \\ 0.500 & 0.866 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}. \tag{2.30}$$

Dado

$$^AP_1 = \begin{bmatrix} 0.0 \\ 2.0 \\ 0.0 \end{bmatrix}, \tag{2.31}$$

calculamos $^AP_2$ como

$$^AP_2 = R_z(30.0)\, ^AP_1 = \begin{bmatrix} -1.000 \\ 1.732 \\ 0.000 \end{bmatrix}. \tag{2.32}$$

Las ecuaciones (2.13) y (2.27) implementan los mismos cálculos. Si hubiéramos definido ^B_AR (en vez de ^A_BR) en (2.13), aparecería entonces el inverso de R en (2.27). Este cambio indicaría la diferencia entre girar el vector "hacia adelante" y girar el sistema de coordenadas "hacia atrás". Al definir la ubicación de $\{B\}$ con respecto a $\{A\}$ (mediante ^A_BR), hacemos que los cálculos de las dos interpretaciones sean iguales.

Operadores de transformación

Al igual que con los vectores y las matrices de rotación, una trama tiene otra interpretación como *operador de transformación*. En esta interpretación sólo se involucra a un sistema de coordenadas, por lo que el símbolo T se utiliza sin subíndices ni superíndices. El operador T gira y traslada un vector ${}^{A}P_1$ para calcular un nuevo vector.

$$ {}^{A}P_2 = T\ {}^{A}P_1. \tag{2.33} $$

De nuevo, como en el caso de las rotaciones, los cálculos descritos en (2.18) y en (2.33) son los mismas, sólo nuestra interpretación es distinta. Este hecho también nos permite ver cómo obtener transformadas homogéneas que se vayan a utilizar como operadores:

La transformada que gira mediante R y se traslada mediante Q es la misma que la transformada que describe a una trama girada mediante R y trasladada mediante Q con respecto a la trama de referencia.

Generalmente se asume que una transformada se encuentra en la forma homogénea con partes generales correspondientes a la matriz de rotación y al vector de posición.

EJEMPLO 2.4

La figura 2.11 muestra a un vector ${}^{A}P_1$. Deseamos girarlo 30 grados sobre $\hat{Z}$ y trasladarlo 10 unidades en $\hat{X}_A$ y 5 unidades en $\hat{Y}_A$. Encuentre ${}^{A}P_2$, en donde ${}^{A}P_1 = [3.0\ 7.0\ 0.0]^T$.

El operador T que realiza la traslación y rotación es

$$ T = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 10.0 \\ 0.500 & 0.866 & 0.000 & 5.0 \\ 0.000 & 0.000 & 1.000 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.34} $$

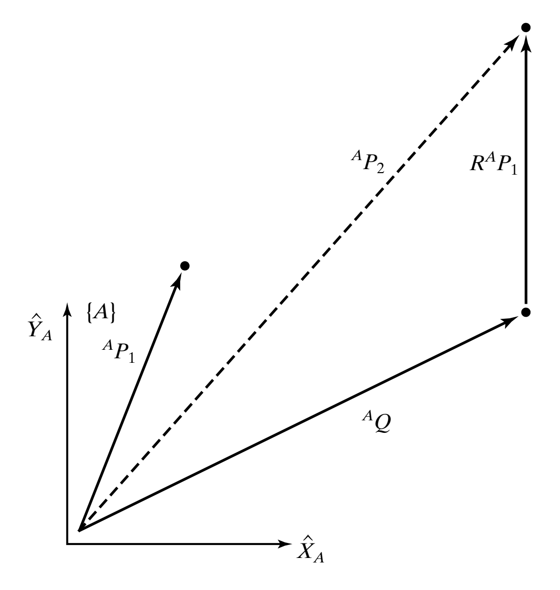


FIGURA 2.11: El vector ${}^{A}P_1$ se gira y traslada para formar ${}^{A}P_2$.

Dado

$$ {}^{A}P_1 = \begin{bmatrix} 3.0 \\ 7.0 \\ 0.0 \end{bmatrix}, \tag{2.35} $$

utilizamos T como un operador:

$$ {}^{A}P_2 = T\ {}^{A}P_1 = \begin{bmatrix} 9.098 \\ 12.562 \\ 0.000 \end{bmatrix}. \tag{2.36} $$

Observe que este ejemplo es, numéricamente idéntico al del ejemplo 2.2, pero la interpretación es bastante distinta.

2.5 RESUMEN DE INTERPRETACIONES

Hemos introducido conceptos en cierto orden: primero para el caso de la traslación solamente, después para el caso de la rotación y finalmente para el caso general de rotación sobre un punto y la traslación de ese punto. Después de haber comprendido el caso general de rotación y traslación no necesitaremos considerar explícitamente los dos casos más simples ya que están contenidos dentro del marco general de trabajo.

Como herramienta general para representar tramas presentamos la *transformada homogénea*, una matriz de 4×4 que contiene información de orientación y posición.

Hemos introducido tres interpretaciones de esta transformada homogénea:

1. Es la *descripción de una trama*. ${}^{A}_{B}T$ describe la trama $\{B\}$ en relación con la trama $\{A\}$. Específicamente, las columnas de ${}^{A}_{B}R$ son vectores unitarios que definen las direcciones de los ejes principales de $\{B\}$, y ${}^{A}P_{BORG}$ ubica la posición del origen de $\{B\}$.
2. Es la *asignación de una transformada*. ${}^{A}_{B}T$ asigna ${}^{B}P \rightarrow {}^{A}P$.
3. Es un *operador de transformación*. T opera en ${}^{A}P_1$ para crear ${}^{A}P_2$.

De aquí en adelante utilizaremos los términos *trama* y *transformada* para referirnos a un vector de posición más una orientación. *Trama* es el término preferido al hablar de una descripción, y *transformada* se utiliza más frecuentemente cuando se implica la función como una asignación u operador. Observe que las transformaciones son generalizaciones de (e incluyen) traslaciones y rotaciones; a menudo utilizaremos el término *transformada* para hablar de una rotación (o traslación) pura.

2.6 ARITMÉTICA DE TRANSFORMACIONES

En esta sección veremos la multiplicación y la inversión de transformadas. Estas dos operaciones elementales forman un conjunto funcionalmente completo de operadores de transformación.

Transformaciones compuestas

En la figura 2.12 tenemos a ${}^{C}P$ y deseamos encontrar ${}^{A}P$.

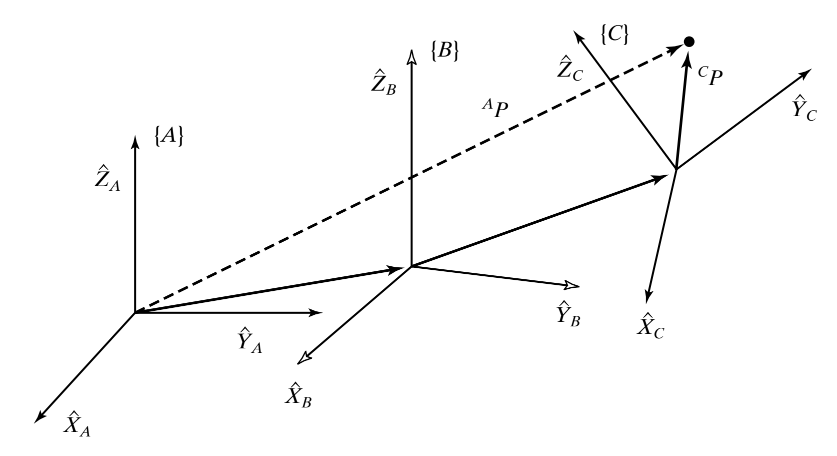


FIGURA 2.12: Tramas compuestas: cada una se conoce en forma relativa a la anterior.

La trama $\{C\}$ se conoce en términos de la trama $\{B\}$, y la trama $\{B\}$ se conoce en términos de la trama $\{A\}$. Podemos transformar a ^{C}P en ^{B}P por medio de

$$^{B}P = {}^{B}_{C}T \; {}^{C}P; \tag{2.37}$$

después podemos transformar ^{B}P en ^{A}P por medio de

$$^{A}P = {}^{A}_{B}T \; {}^{B}P. \tag{2.38}$$

Si combinamos (2.37) y (2.38) obtendremos el siguiente resultado (definitivamente esperado):

$$^{A}P = {}^{A}_{B}T {}^{B}_{C}T {}^{C}P, \tag{2.39}$$

a partir del cual podríamos definir

$${}^{A}_{C}T = {}^{A}_{B}T {}^{B}_{C}T. \tag{2.40}$$

De nuevo, observe que la familiaridad con la notación de subíndices y superíndices hace que estas manipulaciones sean simples. En términos de las descripciones conocidas de $\{B\}$ y $\{C\}$, podemos definir la expresión para ${}^{A}_{C}T$ como:

$${}^{A}_{C}T = \left[\begin{array}{ccc|c} & {}^{A}_{B}R \, {}^{B}_{C}R & & {}^{A}_{B}R \, {}^{B}P_{CORG} + {}^{A}P_{BORG} \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \tag{2.41}$$

Invertir una transformada

Considere una trama $\{B\}$ que se conoce con respecto a la trama $\{A\}$; es decir, conocemos el valor de ${}^{A}_{B}T$. Algunas veces es conveniente invertir esta transformada para obtener una descripción de $\{A\}$ en forma relativa a $\{B\}$; es decir, ${}^{B}_{A}T$. Una manera sencilla de calcular la inversa es calculando el inverso de la transformada homogénea de 4×4. Pero si hacemos esto no estamos aprovechando completamente la estructura inherente en la transformada. Es fácil encontrar un método más sencillo para calcular la inversa, uno que aproveche esta estructura.

Para encontrar ${}^B_AT$ debemos calcular ${}^B_AR$ y ${}^BP_{AORG}$ a partir de ${}^A_BR$ y de ${}^AP_{BORG}$. Recuerde de nuestra discusión sobre matrices de rotación que

$$ {}^B_AR = {}^A_BR^T. \tag{2.42} $$

A continuación cambiamos la descripción de ${}^AP_{BORG}$ a $\{B\}$ utilizando la ecuación (2.13):

$$ {}^B({}^AP_{BORG}) = {}^B_AR\, {}^AP_{BORG} + {}^BP_{AORG}. \tag{2.43} $$

El lado izquierdo de (2.43) debe ser cero, por lo que tenemos que

$$ {}^BP_{AORG} = -{}^B_AR\, {}^AP_{BORG} = -{}^A_BR^T\, {}^AP_{BORG}. \tag{2.44} $$

Utilizando las ecuaciones (2.42) y (2.44), podemos escribir la forma de ${}^B_AT$ como

$$ {}^B_AT = \left[\begin{array}{ccc|c} & {}^A_BR^T & & -{}^A_BR^T\, {}^AP_{BORG} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{2.45} $$

Observe que, con nuestra notación,

$$ {}^B_AT = {}^A_BT^{-1}. $$

La ecuación (2.45) es una manera general y extremadamente útil de calcular la inversa de una transformada homogénea.

EJEMPLO 2.5

La figura 2.13 muestra una trama $\{B\}$ que se gira 30 grados sobre $\hat{Z}$ respecto a la trama $\{A\}$ y se traslada cuatro unidades en $\hat{X}_A$ y tres unidades en $\hat{Y}_A$. De esta manera tenemos una descripción de ${}^A_BT$. Encuentre ${}^B_AT$.

La trama que define $\{B\}$ es

$$ {}^A_BT = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 4.0 \\ 0.500 & 0.866 & 0.000 & 3.0 \\ 0.000 & 0.000 & 1.000 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.46} $$

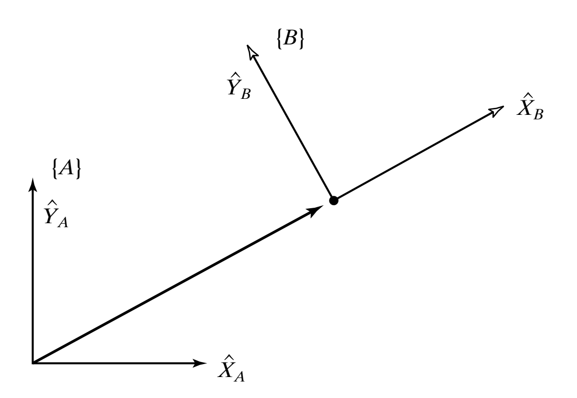


FIGURA 2.13: $\{B\}$ relativa a $\{A\}$.

Utilizando la ecuación (2.45) calculamos

$$ {}^{B}_{A}T = \begin{bmatrix} 0.866 & 0.500 & 0.000 & -4.964 \\ -0.500 & 0.866 & 0.000 & -0.598 \\ 0.000 & 0.000 & 1.000 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.47} $$

2.7 ECUACIONES DE TRANSFORMADAS

La figura 2.14 indica una situación en la que una trama $\{D\}$ puede expresarse como productos de transformaciones en dos formas distintas. Primera forma:

$$ {}^{U}_{D}T = {}^{U}_{A}T \, {}^{A}_{D}T; \tag{2.48} $$

segunda forma:

$$ {}^{U}_{D}T = {}^{U}_{B}T \, {}^{B}_{C}T \, {}^{C}_{D}T. \tag{2.49} $$

Podemos hacer que estas dos descripciones de ${}^{U}_{D}T$ sean iguales para construir una **ecuación de transformadas**:

$$ {}^{U}_{A}T \, {}^{A}_{D}T = {}^{U}_{B}T \, {}^{B}_{C}T \, {}^{C}_{D}T. \tag{2.50} $$

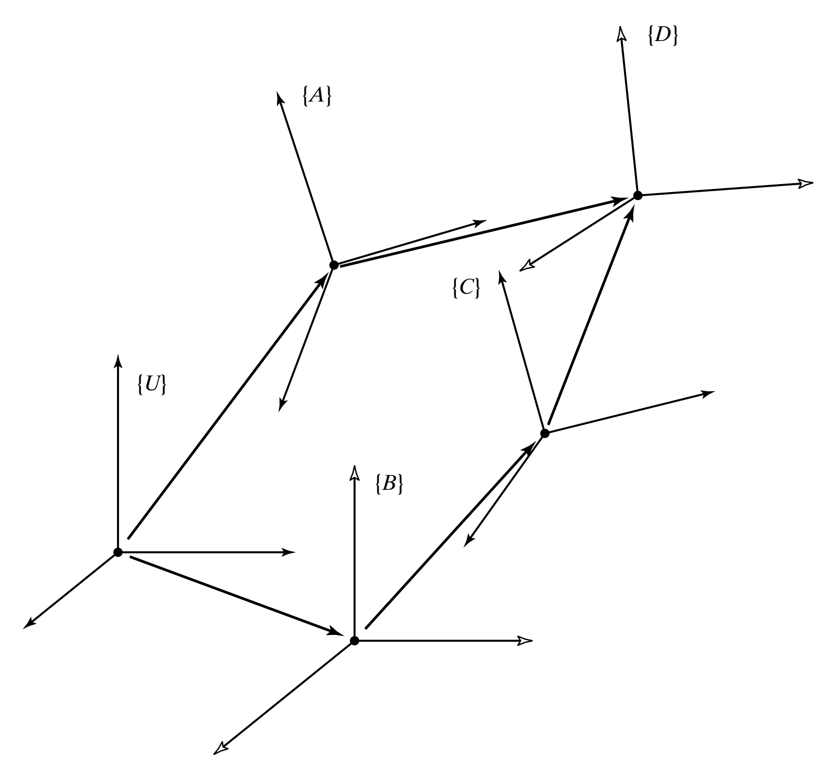


FIGURA 2.14: Conjunto de transformadas que forman un lazo.

Las ecuaciones de transformadas pueden usarse para resolver transformadas en el caso de n transformadas desconocidas y n ecuaciones de transformadas. Considere la ecuación (2.50) en el caso en que todas las transformadas se conocen, excepto ${}^{B}_{C}T$. Aquí tenemos una ecuación de transformadas y una transformada desconocida; por lo tanto, podemos encontrar fácilmente su solución:

$$ {}^{B}_{C}T = {}^{U}_{B}T^{-1}\, {}^{U}_{A}T\, {}^{A}_{D}T\, {}^{C}_{D}T^{-1}. \tag{2.51}$$

La figura 2.15 indica una situación similar.

Observe que en todas las figuras hemos introducido una representación *gráfica* de tramas como una flecha que apunta de un origen a otro. La dirección de la flecha indica de qué manera se definen las tramas. En la figura 2.14 la trama $\{D\}$ se define en forma relativa a $\{A\}$; en la figura 2.15 la trama $\{A\}$ se define en forma relativa a $\{D\}$. Para poder combinar las tramas al alinearse las flechas, simplemente calculamos el producto de las transformadas. Si una flecha apunta en sentido opuesto en una cadena de transformadas, simplemente calculamos primero su inversa. En la figura 2.15, dos posibles descripciones de $\{C\}$ son

$$ {}^{U}_{C}T = {}^{U}_{A}T\, {}^{D}_{A}T^{-1}\, {}^{D}_{C}T \tag{2.52}$$

y

$$ {}^{U}_{C}T = {}^{U}_{B}T\, {}^{B}_{C}T. \tag{2.53}$$

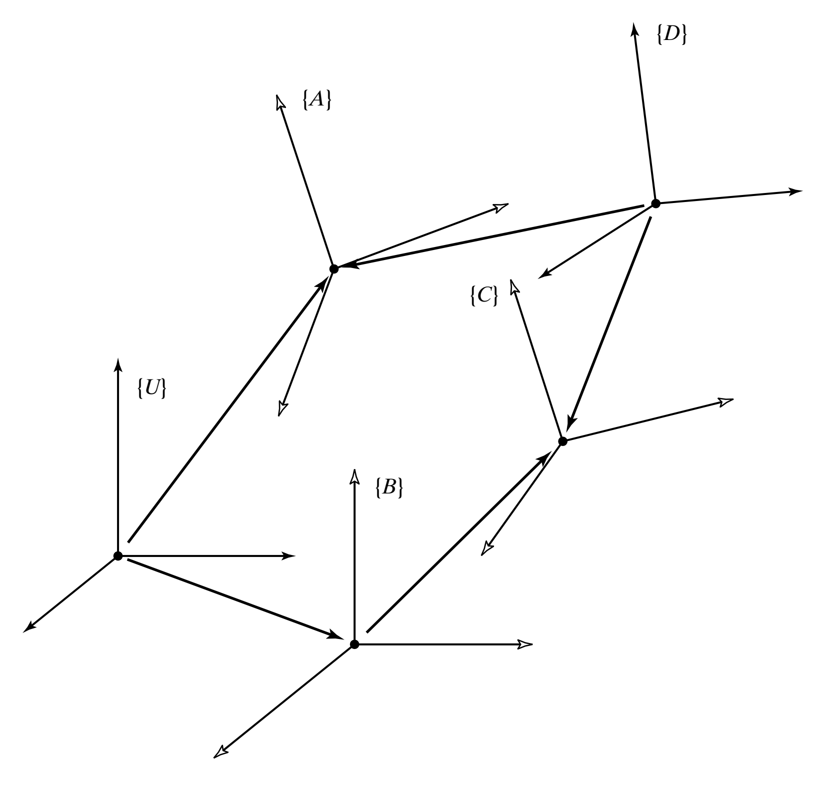


FIGURA 2.15: Ejemplo de una ecuación de transformadas.

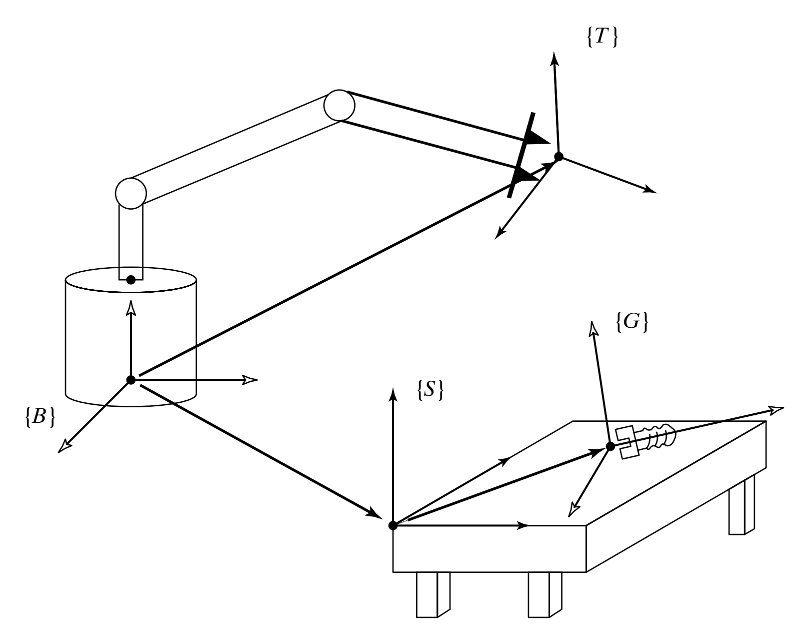


FIGURA 2.16: Un manipulador tratando de agarrar un perno.

De nuevo, podríamos igualar las ecuaciones (2.52) y (2.53) para despejar, por decir, ${}^{U}_{A}T$:

$$ {}^{U}_{A}T = {}^{U}_{B}T\ {}^{B}_{C}T\ {}^{D}_{C}T^{-1}\ {}^{D}_{A}T. \tag{2.54}$$

EJEMPLO 2.6

Suponga que conocemos la transformada ${}^{B}_{T}T$, en la figura 2.16, que describe la trama $\{T\}$ en la punta de los dedos del manipulador, relativa a su base, $\{B\}$; que también sabemos en dónde se ubica la parte superior de la mesa en el espacio, relativa a la base del manipulador (porque tenemos una descripción de la trama $\{S\}$ que está unida a la mesa, como se muestra, ${}^{B}_{S}T$), y que conocemos la ubicación de la trama unida al perno que está sobre la mesa, en forma relativa a la trama de la mesa; es decir, ${}^{S}_{G}T$. Calcule la posición y la orientación del perno, de manera relativa a la mano del manipulador, ${}^{T}_{G}T$.

Si nos guiamos por nuestra notación (y esperamos también que por nuestra comprensión), calculamos la trama del perno relativa a la trama de la mano como

$$ {}^{T}_{G}T = {}^{B}_{T}T^{-1}\ {}^{B}_{S}T\ {}^{S}_{G}T. \tag{2.55}$$

2.8 MÁS SOBRE LA REPRESENTACIÓN DE LA ORIENTACIÓN

Hasta ahora nuestro único medio para representar una orientación es mediante una matriz de rotación de 3×3. Como se ha demostrado, las matrices de rotación son especiales en cuanto a que todas las columnas son mutuamente ortogonales y tienen

magnitud unitaria. Además, veremos que el determinante de una matriz de rotación siempre es igual a +1. Las matrices de rotación pueden llamarse también **matrices ortonormales propias**, en donde "propias" se refiere al hecho de que el determinante es +1 (las matrices ortonormales impropias tienen el determinante de −1).

Es natural preguntar si es posible describir una orientación con menos de nueve números. Un resultado del álgebra lineal (conocido como **fórmula de Cayley para matrices ortonormales** [3]) nos dice que, para cualquier matriz ortonormal propia, R, existe una matriz antisimétrica, S, tal que

$$R = (I_3 - S)^{-1}(I_3 + S), \tag{2.56}$$

en donde I_3 es una matriz unitaria de 3 × 3. Ahora, una matriz antisimétrica (es decir, $S = -S^T$) de dimensión 3 se especifica mediante tres parámetros (s_x, s_y, s_z) como

$$S = \begin{bmatrix} 0 & -s_x & s_y \\ s_x & 0 & -s_x \\ -s_y & s_x & 0 \end{bmatrix}. \tag{2.57}$$

Por lo tanto, una consecuencia inmediata de la fórmula (2.56) es que cualquier matriz de rotación de 3 × 3 puede especificarse mediante sólo tres parámetros.

Evidentemente, no todos los (nueve) elementos de una matriz de rotación son independientes. De hecho, dada una matriz de rotación, R, es fácil anotar las seis dependencias entre los elementos. Imagine a R como una matriz de tres columnas, como se introdujo originalmente:

$$R = [\hat{X}\ \hat{Y}\ \hat{Z}]. \tag{2.58}$$

Como sabemos por la sección 2.2, estos tres vectores son los ejes unitarios de alguna trama, escritos en términos de la trama de referencia. Cada uno es un vector unitario y los tres deben ser mutuamente perpendiculares, por lo que podemos ver que hay seis restricciones en los nueve elementos de la matriz:

$$\begin{aligned} |\hat{X}| &= 1, \\ |\hat{Y}| &= 1, \\ |\hat{Z}| &= 1, \\ \hat{X}\cdot\hat{Y} &= 0, \\ \hat{X}\cdot\hat{Z} &= 0, \\ \hat{Y}\cdot\hat{Z} &= 0. \end{aligned} \tag{2.59}$$

Es natural preguntar si pueden idearse representaciones de la orientación de manera que la representación se especifique *convenientemente* con tres parámetros. En esta sección presentaremos varias representaciones de ese tipo.

Mientras que las traslaciones sobre tres ejes mutuamente perpendiculares son bastante fáciles de visualizar, las rotaciones parecen ser menos intuitivas. Desafortunadamente, a la gente se le dificulta describir y especificar orientaciones en espacio tridimensional. Una de las dificultades radica en que las rotaciones generalmente no se conmutan. Esto es, ${}^A_BR\ {}^B_CR$ no es lo mismo que ${}^B_CR\ {}^A_BR$.

EJEMPLO 2.7

Considere dos rotaciones, una de 30 grados sobre $\hat{Z}$ y otra de 30 grados sobre $\hat{X}$:

$$R_z(30) = \begin{bmatrix} 0.866 & -0.500 & 0.000 \\ 0.500 & 0.866 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix} \tag{2.60}$$

$$R_x(30) = \begin{bmatrix} 1.000 & 0.000 & 0.000 \\ 0.000 & 0.866 & -0.500 \\ 0.000 & 0.500 & 0.866 \end{bmatrix} \tag{2.61}$$

$$R_z(30)R_x(30) = \begin{bmatrix} 0.87 & -0.43 & 0.25 \\ 0.50 & 0.75 & -0.43 \\ 0.00 & 0.50 & 0.87 \end{bmatrix}$$

$$\neq R_x(30)R_z(30) = \begin{bmatrix} 0.87 & -0.50 & 0.00 \\ 0.43 & 0.75 & -0.50 \\ 0.25 & 0.43 & 0.87 \end{bmatrix} \tag{2.62}$$

El hecho de que el orden de las rotaciones sea importante no debe sorprender; lo que es más, se captura en el hecho de que utilizamos matrices para representar rotaciones, ya que la multiplicación de matrices en general no es conmutativa.

Como las rotaciones pueden considerarse ya sea como operadores o como descripciones de orientación, no es sorprendente que distintas representaciones sean preferidas para cada uno de estos usos. Las matrices de rotación son útiles como operadores. Su forma matricial es tal que, cuando se multiplican por un vector, realizan la operación de rotación. No obstante, las matrices de rotación son poco manejables cuando se utilizan para especificar una orientación. A un operador humano en una terminal de computadora que desee introducir la especificación de la orientación deseada de la mano de un robot se le dificultaría mucho introducir una matriz de nueve elementos con columnas ortonormales. Sería más simple una representación que requiriera solamente tres números. En las siguientes secciones introduciremos varias de esas representaciones.

Ángulos fijos X – Y – Z

Un método para describir la orientación de una trama $\{B\}$ es el siguiente:

> Empiece con la trama que sea coincidente con una trama de referencia conocida $\{A\}$. Gire $\{B\}$ primero sobre $\hat{X}_A$ usando un ángulo γ, luego sobre $\hat{Y}_A$ usando un ángulo β y finalmente sobre $\hat{Z}_A$ usando un ángulo α.

Cada una de las tres rotaciones se lleva a cabo sobre un eje en la trama de referencia fija $\{A\}$. A esta convención para especificar una orientación la llamaremos **ángulos fijos X-Y-Z**. La palabra "fijos" se refiere al hecho de que las rotaciones se especifican sobre la trama de referencia fija (es decir, inmóvil) (figura 2.17). Algunas veces esta convención se conoce como **ángulos de balanceo, inclinación y orientación** (en inglés, *roll, pitch, yaw*), pero hay que tener cuidado ya que a menudo este nombre se da a otras convenciones, relacionadas pero distintas.

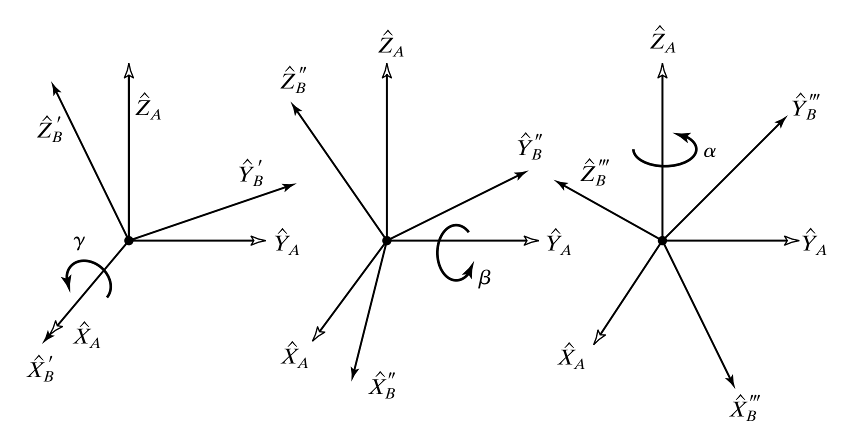


FIGURA 2.17: Ángulos fijos X-Y-Z. Las rotaciones se llevan a cabo en el orden $R_X(\gamma)$, $R_Y(\beta)$, $R_Z(\alpha)$.

La derivación de la matriz de rotación equivalente, ${}^A_BR_{XYZ}(\gamma, \beta, \alpha)$ es directa, ya que todas las rotaciones ocurren sobre ejes de la trama de referencia; esto es:

$$\begin{aligned} {}^A_BR_{XYZ}(\gamma, \beta, \alpha) &= R_Z(\alpha)R_Y(\beta)R_X(\gamma) \\ &= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix}, \end{aligned} \tag{2.63}$$

en donde $c\alpha$ es la abreviación de cos α, $s\alpha$ de sen α y así sucesivamente. Es extremadamente importante comprender el orden de las rotaciones usadas en la ecuación (2.63). Si pensamos en términos de las rotaciones como si fueran operadores, hemos aplicado las rotaciones (a partir de la *derecha*) de $R_X(\gamma)$, después $R_Y(\beta)$ y luego $R_Z(\alpha)$. Después de realizar la multiplicación en (2.63) obtenemos

$${}^A_BR_{XYZ}(\gamma, \beta, \alpha) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}. \tag{2.64}$$

Recuerde que la definición dada aquí especifica el orden de las tres rotaciones. La ecuación (2.64) es correcta sólo para las rotaciones realizadas en el siguiente orden: γ grados sobre $\hat{X}_A$, β grados sobre $\hat{Y}_A$, α grados sobre $\hat{Z}_A$.

El problema inverso, el de extraer los ángulos fijos X-Y-Z equivalentes a partir de una matriz de rotación, es a menudo de interés. La solución depende de resolver un conjunto de ecuaciones trascendentales: hay nueve ecuaciones y tres incógnitas si (2.64) se iguala a una matriz de rotación dada. Entre las nueve ecuaciones hay seis dependencias, por lo que, esencialmente, tenemos tres ecuaciones y tres incógnitas. Supongamos que

$${}^A_BR_{XYZ}(\gamma, \beta, \alpha) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \tag{2.65}$$

De la ecuación (2.64) podemos ver que, al sacar la raíz cuadrada de la suma de los cuadrados de r_{11} y r_{21}, podemos calcular cos β. Después podemos resolver para β con

el arco tangente de $-r_{31}$ sobre el coseno calculado. Luego, siempre y cuando $c\beta \neq 0$, podemos resolver para α sacando el arco tangente de $r_{21}/c\beta$ sobre $r_{11}/c\beta$ y podemos resolver para γ sacando el arco tangente de $r_{32}/c\beta$ sobre $r_{33}/c\beta$.

En resumen:

$$\begin{aligned} \beta &= \text{Atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}), \\ \alpha &= \text{Atan2}(r_{21}/c\beta, r_{11}/c\beta), \\ \gamma &= \text{Atan2}(r_{32}/c\beta, r_{33}/c\beta), \end{aligned} \tag{2.66}$$

en donde Atan2(y, x) es una función arco tangente de dos argumentos.[3]

Aunque existe una segunda solución, al utilizar la raíz cuadrada negativa en la fórmula para β siempre calculamos la única solución para la cual $-90.0° \leq \beta \leq 90.0°$. Esto es recomendable ya que podemos definir funciones de asignación de uno a uno entre varias representaciones de la orientación. No obstante, en algunos casos es importante calcular todas las soluciones (profundizaremos en este tema en el capítulo 4).

Si $\beta = \pm 90.0°$ (para que $c\beta = 0$), la solución de (2.66) se degenera. En esos casos sólo puede calcularse la suma o diferencia de α y γ. Una posible convención es elegir $\alpha = 0$ en estos casos, lo cual nos da los resultados que se muestran a continuación.

Si $\beta = 90.0°$, puede calcularse una solución para que

$$\begin{aligned} \beta &= 90.0°, \\ \alpha &= 0.0, \\ \gamma &= \text{Atan2}(r_{12}, r_{22}). \end{aligned} \tag{2.67}$$

Si $\beta = -90.0°$, puede calcularse una solución para que

$$\begin{aligned} \beta &= -90.0°, \\ \alpha &= 0.0, \\ \gamma &= -\text{Atan2}(r_{12}, r_{22}). \end{aligned} \tag{2.68}$$

Ángulos de Euler Z – Y – X

Otra posible descripción de una trama $\{B\}$ podría ser la siguiente:

> Comience con la trama que sea coincidente con una trama $\{A\}$ conocida. Gire $\{B\}$ primero sobre $\hat{Z}_B$ usando un ángulo α, después sobre $\hat{Y}_B$, usando un ángulo β, y finalmente sobre $\hat{X}_B$ usando un ángulo γ.

En esta representación, cada rotación se lleva a cabo sobre un eje del sistema $\{B\}$ móvil, en lugar de hacerlo sobre uno de la referencia fija $\{A\}$. Dichos conjuntos de tres

[3] Atan2(y, x) calcula $\tan^{-1}\left(\frac{y}{x}\right)$ pero utiliza los signos tanto de x como de y para identificar el cuadrante en el que se encuentra el ángulo resultante. Por ejemplo, Atan2(-2.0, -2.0) $= -135°$, mientras que Atan2(2.0, 2.0) $= 45°$, una distinción que se perdería con una función arco tangente de un solo argumento. Frecuentemente estamos calculando ángulos que pueden variar sobre un giro completo de 360°, por lo que utilizaremos comúnmente la función Atan2. Observe que Atan2 se vuelve indefinida cuando ambos argumentos son iguales a cero. Algunas veces se le llama "arco tangente de 4 cuadrantes" y algunas bibliotecas de lenguajes de programación la tienen predefinida.

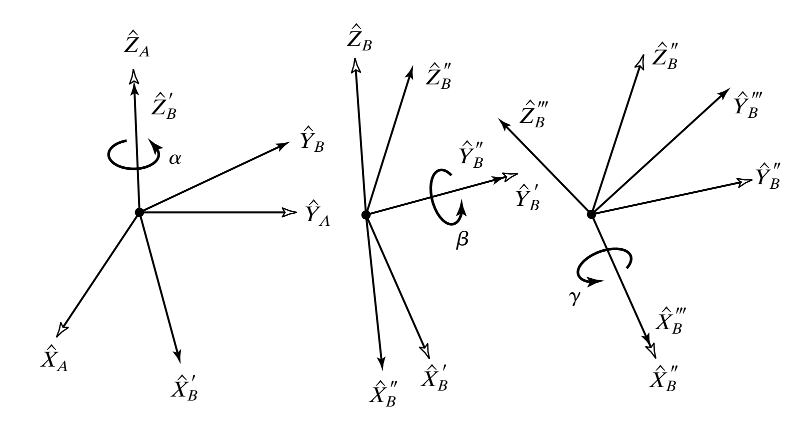


FIGURA 2.18: Ángulos Z-Y-X de Euler.

rotaciones se llaman **ángulos de Euler**. Observe que cada rotación se lleva a cabo sobre un eje cuya ubicación depende de las rotaciones anteriores. Como las tres rotaciones ocurren sobre los ejes $\hat{Z}$, $\hat{Y}$ y $\hat{X}$, llamaremos a esta representación **ángulos de Euler Z-Y-X**.

La figura 2.18 muestra los ejes de $\{B\}$ después de aplicar cada rotación por un ángulo de Euler. La rotación α sobre $\hat{Z}$ ocasiona que X gire hacia $\hat{X}'$, que $\hat{Y}$ gire hacia $\hat{Y}'$ y así sucesivamente. Se agrega una "prima" adicional a cada eje con cada rotación. Una matriz de rotación que esté parametrizada por ángulos de Euler Z-Y-X se indicará mediante la notación ${}^A_B R_{Z'Y'X'}(\alpha, \beta, \gamma)$. Observe que hemos agregado "primas" a los subíndices para indicar que esta rotación es descrita por ángulos de Euler.

Con referencia a la figura 2.18, podemos usar las tramas intermedias $\{B'\}$ y $\{B''\}$ para poder obtener una expresión para ${}^A_B R_{Z'Y'X'}(\alpha, \beta, \gamma)$. Pensando en las rotaciones como las descripciones de estas tramas, podemos escribir inmediatamente

$$ {}^A_B R = {}^A_{B'}R \; {}^{B'}_{B''}R \; {}^{B''}_{B}R, \tag{2.69} $$

en donde cada una de las descripciones relativas en el lado derecho de (2.69) se da por el enunciado de la convención del ángulo de Euler Z-Y-X. En sí, la orientación final de $\{B\}$ se da en forma relativa a $\{A\}$ como

$$ \begin{aligned} {}^A_B R_{Z'Y'X'} &= R_Z(\alpha) R_Y(\beta) R_X(\gamma) \\ &= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix}, \end{aligned} \tag{2.70} $$

en donde $c\alpha = \cos\alpha$, $s\alpha = \operatorname{sen}\alpha$, y así sucesivamente. Al multiplicar obtenemos

$$ {}^A_B R_{Z'Y'X'}(\alpha, \beta, \gamma) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}. \tag{2.71} $$

Observe que el resultado es exactamente el mismo que se obtiene para las mismas tres rotaciones que se realizan en el orden opuesto sobre ejes fijos. Este no muy intuitivo

resultado tiene validez general: tres rotaciones realizadas sobre tres ejes fijos producen igual orientación final que las mismas tres rotaciones realizadas en orden opuesto sobre los ejes de la trama móvil.

Como la ecuación (2.71) es equivalente a la ecuación (2.64), no hay necesidad de repetir la solución para extraer ángulos de Euler Z-Y-X a partir de una matriz de rotación. Esto es, la ecuación (2.66) puede usarse también para resolver ángulos de Euler Z-Y-X que correspondan a una matriz de rotación dada.

Ángulos de Euler Z-Y-Z

Otra posible descripción de una trama $\{B\}$ es

> Empiece con la trama que sea coincidente con una trama $\{A\}$ conocida. Gire $\{B\}$ primero sobre Z_B usando un ángulo α, después sobre Y_B usando un ángulo β y, finalmente, sobre Z_b con un ángulo γ.

Las rotaciones se describen en forma relativa a la trama que estamos moviendo, en este caso $\{B\}$, por lo que ésta es una descripción de ángulo de Euler. Como las tres rotaciones ocurren sobre los ejes $\hat{Z}$, Y y $\hat{Z}$, llamaremos a esta representación **ángulos de Euler Z-Y-Z**.

Siguiendo exactamente el mismo desarrollo que en la sección anterior, llegamos a la matriz de rotación equivalente

$$ {}^{A}_{B}R_{Z'Y'Z'}(\alpha,\beta,\gamma) = \begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix}. \tag{2.72} $$

La solución para extraer ángulos Z-Y-Z de Euler a partir de una matriz de rotación se da a continuación.

Dada:

$$ {}^{A}_{B}R_{Z'Y'Z'}(\alpha,\beta,\gamma) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \tag{2.73} $$

Entonces, si sen $\beta \neq 0$, resulta que

$$ \begin{aligned} \beta &= \text{Atan2}(\sqrt{r_{31}^2 + r_{32}^2}, r_{33}), \\ \alpha &= \text{Atan2}(r_{23}/s\beta, r_{13}/s\beta), \\ \gamma &= \text{Atan2}(r_{32}/s\beta, -r_{31}/s\beta). \end{aligned} \tag{2.74} $$

Aunque existe una segunda solución (que encontramos utilizando la raíz cuadrada negativa en la fórmula para β), siempre calculamos la única solución para la cual $0.0° \leq \beta \leq 180.0°$. Si $\beta = 0.0$ o $180.0°$, la solución de (2.74) se degenera. En esos casos sólo puede calcularse la suma o la diferencia de α y de γ. Una posible convención es elegir $\alpha = 0.0$ en estos casos, lo cual nos da los siguientes resultados.

Si $\beta = 0.0$, entonces puede calcularse una solución para que

$$ \begin{aligned} \beta &= 0.0, \\ \alpha &= 0.0, \\ \gamma &= \text{Atan2}(-r_{12}, r_{11}). \end{aligned} \tag{2.75} $$

Si $\beta = 180.0°$, entonces puede calcularse una solución para que

$$\begin{aligned} \beta &= 180.0°, \\ \alpha &= 0.0, \\ \gamma &= \text{Atan2}(r_{12}, -r_{11}). \end{aligned} \tag{2.76}$$

Otras convenciones de fijación de ángulos

En las subsecciones anteriores hemos visto tres convenciones para especificar la orientación: ángulos fijos X-Y-Z, ángulos de Euler Z-Y-X y ángulos de Euler Z-Y-Z. Cada una de estas convenciones requiere de la realización de tres rotaciones sobre los ejes principales en un cierto orden y son ejemplos de un conjunto de 24 convenciones que llamaremos **convenciones de fijación de ángulos**. De éstas, 12 son para conjuntos de ángulos fijos y 12 para conjuntos de ángulos de Euler. Observe que, debido a la dualidad de los conjuntos de ángulos fijos con los conjuntos de ángulos de Euler, realmente hay sólo 12 parametrizaciones únicas de una matriz de rotación utilizando rotaciones sucesivas sobre los ejes principales. A menudo no hay una razón específica para preferir una convención sobre otra, pero como varios autores adoptan distintas convenciones es útil tener una lista de las matrices de rotación equivalentes para las 24 convenciones. En el apéndice B de este libro se proporcionan las matrices de rotación equivalentes para las 24 convenciones.

Representación de ángulo-eje equivalente

Con la notación $R_X(30.0)$ damos la descripción de una orientación al proporcionar un eje $\hat{X}$ y un ángulo de 30.0 grados, que es un ejemplo de una representación de **ángulo-eje equivalente**. Si el eje es una dirección *general* (en vez de una de las direcciones unitarias) puede obtenerse cualquier orientación a través de la selección apropiada de eje y ángulo. Considere la siguiente descripción de una trama $\{B\}$:

> Empiece con la trama que sea coincidente con una trama $\{A\}$ conocida; después gire $\{B\}$ sobre el vector ${}^A\hat{K}$ usando un ángulo θ de acuerdo con la regla de la mano derecha.

Al vector $\hat{K}$ algunas veces se le llama eje equivalente de una rotación finita. Una orientación general de $\{B\}$ relativa a $\{A\}$ puede escribirse como ${}^A_BR(\hat{K}, \theta)$ o $R_K(\theta)$ y se le llama representación de ángulo-eje equivalente.[4] La especificación del vector ${}^A\hat{K}$ requiere sólo de dos parámetros, ya que su longitud siempre se considera como igual a uno. El ángulo especifica un tercer parámetro. A menudo multiplicamos la dirección unitaria, $\hat{K}$, con la cantidad de rotación, θ, para formar una descripción de orientación mediante un vector compacto de 3×1, denotado por K (sin "gorro"). Vea la figura 2.19.

Cuando se selecciona el eje de rotación de entre los ejes principales de $\{A\}$, la matriz de rotación equivalente toma la forma familiar de rotaciones planares:

$$R_X(\theta) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & \cos\theta & -\operatorname{sen}\theta \\ 0 & \operatorname{sen}\theta & \cos\theta \end{bmatrix}, \tag{2.77}$$

[4] El que existan $\hat{K}$ y θ para cualquier orientación de $\{B\}$ relativa a $\{A\}$ lo demostró originalmente Euler; a esto se le conoce como Teorema de Euler sobre rotación [3].

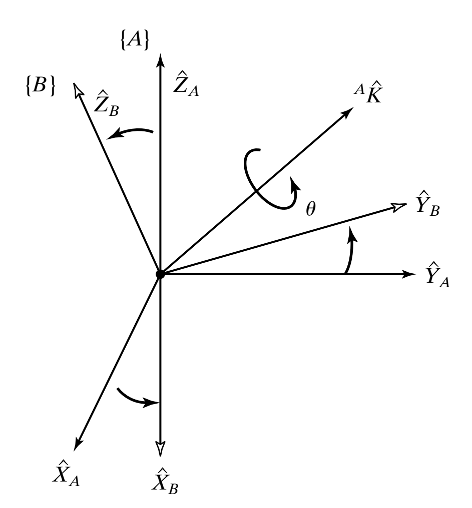


FIGURA 2.19: Representación de ángulo-eje equivalente.

$$R_Y(\theta) = \begin{bmatrix} \cos\theta & 0 & \operatorname{sen}\theta \\ 0 & 1 & 0 \\ -\operatorname{sen}\theta & 0 & \cos\theta \end{bmatrix}, \tag{2.78}$$

$$R_Z(\theta) = \begin{bmatrix} \cos\theta & -\operatorname{sen}\theta & 0 \\ \operatorname{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.79}$$

Si el eje de rotación es un eje arbitrario, puede mostrarse (como en el ejercicio 2.6) que la matriz de rotación equivalente es

$$R_K(\theta) = \begin{bmatrix} k_x k_x v\theta + c\theta & k_x k_y v\theta - k_z s\theta & k_x k_z v\theta + k_y s\theta \\ k_x k_y v\theta + k_z s\theta & k_y k_y v\theta + c\theta & k_y k_z v\theta - k_x s\theta \\ k_x k_z v\theta - k_y s\theta & k_y k_z v\theta + k_x s\theta & k_z k_z v\theta + c\theta \end{bmatrix}, \tag{2.80}$$

en donde $c\theta = \cos\theta$, $s\theta = \operatorname{sen}\theta$, $v\theta = 1 - \cos\theta$ y ${}^A\hat{K} = [k_x k_y k_z]^T$. El signo de θ se determina mediante la regla de la mano derecha, en donde el pulgar apunta hacia el sentido positivo de ${}^A\hat{K}$.

La ecuación (2.80) convierte de representación de ángulo-eje a representación de matriz de rotación. Observe que, dado cualquier eje de rotación y cualquier cantidad angular, podemos construir fácilmente una matriz de rotación equivalente.

El problema inverso de calcular $\hat{K}$ y θ a partir de una matriz de rotación dada se deja en su mayor parte para los ejercicios (ejercicios 2.6 y 2.7), aunque enseguida mostramos un resultado parcial [3]. Si

$${}^A_B R_K(\theta) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \tag{2.81}$$

entonces

$$\theta = \operatorname{Acos}\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right)$$

y

$$\hat{K} = \frac{1}{2\,\text{sen}\,\theta}\begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}. \tag{2.82}$$

Esta solución siempre calcula un valor de θ entre 0 y 180 grados. Para cualquier par de ángulo-eje $({}^{A}K, \theta)$ hay otro par $(-{}^{A}K, -\theta)$ que resulta en la misma orientación en el espacio y lo describe la misma matriz de rotación. Por lo tanto, al convertir de una matriz de rotación en una representación de ángulo-eje, nos encontramos con que tenemos que elegir una de varias soluciones. Un problema más grave es que, para rotaciones angulares pequeñas, el eje se vuelve mal definido. Evidentemente, si la cantidad de rotación se vuelve cero, el eje de rotación se vuelve completamente indefinido. La solución dada por la ecuación (2.82) falla si $\theta = 0°$ o si $\theta = 180°$.

EJEMPLO 2.8

Una trama $\{B\}$ se describe como inicialmente coincidente con $\{A\}$. Después giramos $\{B\}$ sobre el vector ${}^{A}K = [0.707\ 0.707\ 0.0]^{T}$ (pasando a través del origen) usando un ángulo $\theta = 30$ grados. Proporcione la descripción de la trama $\{B\}$.

Al sustituir en (2.80) se produce la parte relacionada con la matriz de rotación de la descripción de la trama. No hubo traslación del origen, por lo que el vector de posición es $[0, 0, 0]^{T}$. Por lo tanto,

$${}^{A}_{B}T = \begin{bmatrix} 0.933 & 0.067 & 0.354 & 0.0 \\ 0.067 & 0.933 & -0.354 & 0.0 \\ -0.354 & 0.354 & 0.866 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}. \tag{2.83}$$

Hasta este punto, todas las rotaciones que hemos visto han sido sobre ejes que pasan a través del origen del sistema de referencia. Si encontramos un problema para el que esto no sea cierto, podemos reducir el problema al caso del "eje a través del origen" definiendo tramas adicionales cuyos orígenes estén sobre el eje y luego resolviendo una ecuación de transformadas.

EJEMPLO 2.9

Una trama $\{B\}$ se describe como inicialmente coincidente con $\{A\}$. Después giramos $\{B\}$ sobre el vector ${}^{A}K = [0.707\ 0.707\ 0.0]^{T}$ (pasando por el punto ${}^{A}P = [1.0\ 2.0\ 3.0]$) usando un ángulo $\theta = 30$ grados. Proporcione la descripción de la trama $\{B\}$.

Antes de la rotación, $\{A\}$ y $\{B\}$ son coincidentes. Como se muestra en la figura 2.20, definimos dos nuevas tramas $\{A'\}$ y $\{B'\}$, las cuales son coincidentes entre sí y tienen la misma orientación que $\{A\}$ y $\{B\}$ respectivamente, pero se trasladan en forma relativa a $\{A\}$ por un desplazamiento que coloca sus orígenes en el eje de rotación. Seleccionaremos

$${}^{A}_{A'}T = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 1.0 & 0.0 & 2.0 \\ 0.0 & 0.0 & 1.0 & 3.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}. \tag{2.84}$$

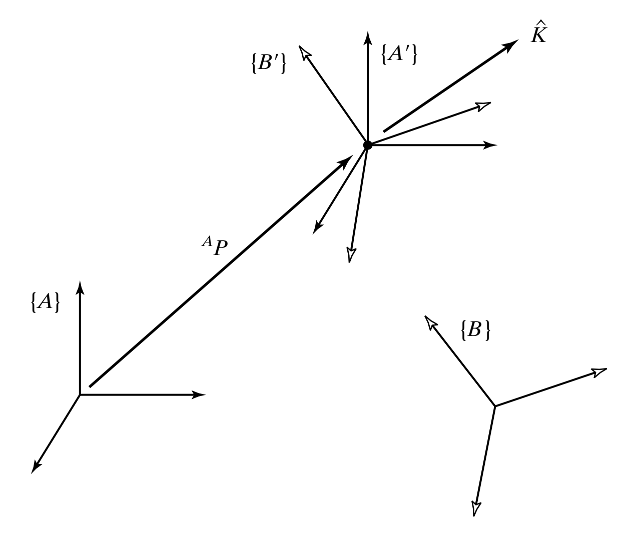


FIGURA 2.20: Rotación sobre un eje que no pasa a través del origen de $\{A\}$. Inicialmente $\{B\}$ era coincidente con $\{A\}$.

De manera similar, la descripción de $\{B\}$ en términos de $\{B'\}$ es:

$$ {}^{B'}_{B}T = \begin{bmatrix} 1.0 & 0.0 & 0.0 & -1.0 \\ 0.0 & 1.0 & 0.0 & -2.0 \\ 0.0 & 0.0 & 1.0 & -3.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}. \tag{2.85} $$

Ahora, si mantenemos las otras relaciones fijas, podemos girar $\{B'\}$ con respecto a $\{A'\}$. Ésta es una rotación sobre un eje que pasa a través del origen, por lo que podemos usar (2.80) para calcular $\{B'\}$ relativa a $\{A'\}$. Al sustituir en (2.80) se produce la parte correspondiente a la matriz de rotación de la descripción de la trama. No hubo traslación del origen, por lo que el vector de posición es $[0, 0, 0]^T$. Por lo tanto, tenemos que

$$ {}^{A'}_{B'}T = \begin{bmatrix} 0.933 & 0.067 & 0.354 & 0.0 \\ 0.067 & 0.933 & -0.354 & 0.0 \\ -0.354 & 0.354 & 0.866 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}. \tag{2.86} $$

Finalmente podemos escribir una ecuación transformada para calcular la trama deseada:

$$ {}^{A}_{B}T = {}^{A}_{A'}T \; {}^{A'}_{B'}T \; {}^{B'}_{B}T, \tag{2.87} $$

la cual se evalúa en

$$ {}^{A}_{B}T = \begin{bmatrix} 0.933 & 0.067 & 0.354 & -1.13 \\ 0.067 & 0.933 & -0.354 & 1.13 \\ -0.354 & 0.354 & 0.866 & 0.05 \\ 0.000 & 0.000 & 0.000 & 1.00 \end{bmatrix}. \tag{2.88} $$

Una rotación sobre un eje que no pasa a través del origen ocasiona un cambio en la posición, además de la misma orientación final, como si el eje hubiera pasado a través del origen.

Observe que podríamos haber usado cualquier definición de $\{A'\}$ y $\{B'\}$ tal que sus orígenes estuvieran en el eje de rotación. Nuestra elección particular de orientación fue arbitraria, y nuestra elección de la posición del origen fue una entre una infinidad de posibles elecciones que están sobre el eje de rotación. (Vea también el ejercicio 2.14).

Parámetros de Euler

Otra representación de la orientación es mediante cuatro números conocidos como los **parámetros de Euler**. Aunque su completa discusión está más allá del alcance de este libro, mencionaremos aquí la convención como referencia.

En términos del eje equivalente $\hat{K} = [k_x k_y k_z]^T$ y del ángulo equivalente θ, los parámetros de Euler se dan de la siguiente manera:

$$\begin{aligned} \epsilon_1 &= k_x \operatorname{sen} \frac{\theta}{2}, \\ \epsilon_2 &= k_y \operatorname{sen} \frac{\theta}{2}, \\ \epsilon_3 &= k_z \operatorname{sen} \frac{\theta}{2}, \\ \epsilon_4 &= \cos \frac{\theta}{2}. \end{aligned} \tag{2.89}$$

Queda claro que estas cuatro cantidades no son independientes:

$$\epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 + \epsilon_4^2 = 1 \tag{2.90}$$

siempre debe cumplirse. Por lo tanto, una orientación podría visualizarse como un punto en una hiperesfera unitaria en un espacio de cuatro dimensiones.

Algunas veces los parámetros de Euler se ven como un vector de 3×1 más un escalar. No obstante, como un vector de 4×1 los parámetros de Euler se conocen como un **cuaternión unitario**.

La matriz de rotación R_ϵ que es equivalente a un conjunto de parámetros de Euler es:

$$R_\epsilon = \begin{bmatrix} 1 - 2\epsilon_2^2 - 2\epsilon_3^2 & 2(\epsilon_1\epsilon_2 - \epsilon_3\epsilon_4) & 2(\epsilon_1\epsilon_3 + \epsilon_2\epsilon_4) \\ 2(\epsilon_1\epsilon_2 + \epsilon_3\epsilon_4) & 1 - 2\epsilon_1^2 - 2\epsilon_3^2 & 2(\epsilon_2\epsilon_3 - \epsilon_1\epsilon_4) \\ 2(\epsilon_1\epsilon_3 - \epsilon_2\epsilon_4) & 2(\epsilon_2\epsilon_3 + \epsilon_1\epsilon_4) & 1 - 2\epsilon_1^2 - 2\epsilon_2^2 \end{bmatrix}. \tag{2.91}$$

Dada una matriz de rotación, los parámetros de Euler equivalentes son:

$$\begin{aligned} \epsilon_1 &= \frac{r_{32} - r_{23}}{4\epsilon_4}, \\ \epsilon_2 &= \frac{r_{13} - r_{31}}{4\epsilon_4}, \\ \epsilon_3 &= \frac{r_{21} - r_{12}}{4\epsilon_4}, \\ \epsilon_4 &= \frac{1}{2}\sqrt{1 + r_{11} + r_{22} + r_{33}}. \end{aligned} \tag{2.92}$$

Observe que la ecuación (2.92) no es útil en un sentido computacional si la matriz de rotación representa una rotación de 180 grados sobre cierto eje, ya que ϵ_4 tiende a cero. No obstante puede demostrarse que en el límite todas las expresiones en (2.92) siguen siendo finitas, incluso para este caso. De hecho, por las definiciones de la ecuación (2.89) está claro que todas las ϵ_i permanecen en el intervalo $[-1, 1]$.

Orientaciones enseñadas y predefinidas

En muchos sistemas de robots será posible "enseñar" posiciones y orientaciones utilizando el mismo robot. El manipulador se desplaza a una ubicación deseada y se registra esta posición. Una trama enseñada de esta manera no necesariamente tiene que ser una a la que se haga regresar al robot; podría ser la ubicación de una pieza o de un dispositivo. En otras palabras, el robot se utiliza como herramienta de medición que tiene seis grados de libertad. Enseñar una orientación como ésta elimina por completo la necesidad de que el programador humano tenga que tratar con la representación de la orientación. En la computadora, el punto enseñado se almacena como una matriz de rotación (o de cualquier otra forma), pero el usuario no tiene que verlo ni entenderlo. Por lo tanto, se recomiendan ampliamente los sistemas de robots que permiten la enseñanza de tramas mediante el uso del mismo robot.

Además de enseñar tramas, algunos sistemas tienen un conjunto de orientaciones predefinidas, tales como "apuntar hacia abajo" o "apuntar a la izquierda". Para los humanos es muy fácil tratar con estas especificaciones. No obstante, si éste fuera el único medio para describir y especificar la orientación, el sistema estaría muy limitado.

2.9 TRANSFORMACIÓN DE VECTORES LIBRES

En este capítulo nos hemos preocupado principalmente por los vectores de posición. En capítulos posteriores hablaremos también de los vectores de velocidad y fuerza. Estos vectores se transformarán de manera distinta, ya que son un *tipo* diferente de vector.

En mecánica se hace la distinción entre la igualdad y la equivalencia de vectores. *Dos vectores son iguales si tienen las mismas dimensiones, magnitud y dirección*. Dos vectores que se consideran *iguales* podrían tener distintas líneas de acción; por ejemplo, los tres vectores iguales de la figura 2.21. Estos vectores de velocidad tienen las mismas dimensiones, magnitud y dirección, y por lo tanto son iguales de acuerdo con nuestra definición.

Dos vectores son equivalentes en cierta capacidad si cada uno produce el mismo efecto en esa capacidad. Por lo tanto, si el criterio en la figura 2.21 es la distancia recorrida, los tres vectores dan el mismo resultado y son, por lo tanto, equivalentes en esta capacidad. Si el criterio es la altura por encima del plano *xy*, entonces los vectores no son equivalentes a pesar de su igualdad. Por ende, las relaciones entre vectores y las nociones de equivalencia *dependen completamente de la situación que esté ocurriendo*. Lo que es más, vectores que no son iguales podrían causar efectos equivalentes en ciertos casos.

Enseguida definiremos dos clases básicas de cantidades vectoriales que podrían ser útiles.

El término **vector línea** se refiere a un vector que es dependiente de su **línea de acción**, junto con la dirección y la magnitud, para ocasionar sus efectos. A menudo los efectos de un vector de fuerza dependen de su línea de acción (o punto de aplicación), por lo que entonces se consideraría como un vector línea.

Un **vector libre** se refiere a un vector que puede posicionarse en cualquier parte en el espacio sin perder ni cambiar su significado, teniendo en cuenta que se preserven su magnitud y dirección.

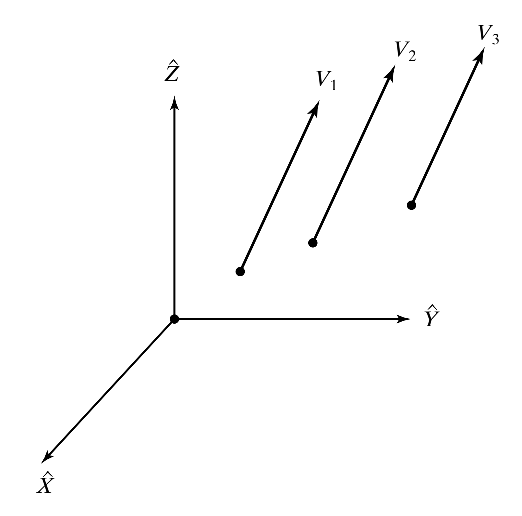


FIGURA 2.21: Vectores con igual velocidad.

Por ejemplo, un vector de momento puro es siempre un vector libre. Si tenemos un vector de momento ${}^{B}N$ que se conoce en términos de $\{B\}$, entonces calculamos el mismo momento en términos de la trama $\{A\}$ como

$$ {}^{A}N = {}^{A}_{B}R \, {}^{B}N. \tag{2.93} $$

En otras palabras, todo lo que cuenta es la magnitud y la dirección (en el caso de un vector libre), por lo que sólo se utiliza la matriz de rotación que relaciona a los dos sistemas en la transformación. Las ubicaciones relativas de los orígenes no entran en el cálculo.

De igual forma, un vector de velocidad escrito en $\{B\}$, ${}^{B}V$, se escribe en $\{A\}$ como

$$ {}^{A}V = {}^{A}_{B}R \, {}^{B}V. \tag{2.94} $$

La velocidad de un punto es un vector libre, por lo que todo lo que importa es su dirección y su magnitud. La operación de rotación, como en la ecuación (2.94), no afecta a la magnitud, pero logra la rotación que cambia la descripción del vector de $\{B\}$ a $\{A\}$. Observe que ${}^{A}P_{BORG}$, que aparecería en una transformación de vector de posición, no aparece en una transformada de velocidad. Por ejemplo, en la figura 2.22, si ${}^{B}V = 5\hat{X}$, entonces ${}^{A}V = 5\hat{Y}$.

Los vectores de velocidad y fuerza, junto con los vectores de momento, se presentarán con más detalle en el capítulo 5.

2.10 CONSIDERACIONES COMPUTACIONALES

La disponibilidad de poder computacional de bajo costo es en gran parte responsable del crecimiento de la industria de la robótica; a pesar de ello, y por mucho tiempo todavía, la computación eficiente seguirá siendo una cuestión importante en el diseño de un sistema de manipulación.

La representación homogénea es útil como entidad conceptual, pero el software de transformación que se utiliza comúnmente en los sistemas de manipulación industriales no hace uso de ella directamente, ya que el tiempo que se invierte multiplicando

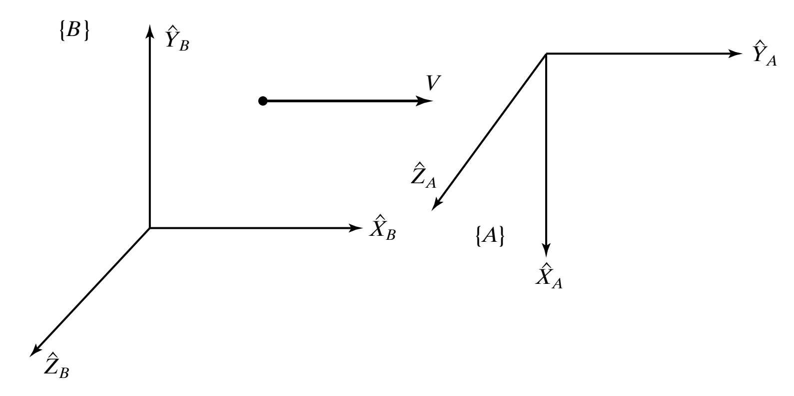


FIGURA 2.22: Transformación de velocidades.

por ceros y unos es un desperdicio. Generalmente, se realizan los cálculos que se muestran en las ecuaciones (2.41) y (2.45), en lugar de la multiplicación directa o inversión de matrices de 4×4.

El *orden* en el que se aplican las transformaciones puede hacer una gran diferencia en la cantidad de cómputo requerido para calcular la misma cantidad. Considere el proceso de realizar múltiples rotaciones de un vector, como en

$$^{A}P = {}^{A}_{B}R\, {}^{B}_{C}R\, {}^{C}_{D}R\, {}^{D}P. \tag{2.95}$$

Una opción es multiplicar primero las tres matrices de rotación para formar ${}^{A}_{D}R$ en la expresión

$$^{A}P = {}^{A}_{D}R\, {}^{D}P. \tag{2.96}$$

Para formar ${}^{A}_{D}R$ a partir de sus tres constituyente se requieren 54 multiplicaciones y 36 sumas. Para realizar la multiplicación final de matriz-vector de la ecuación (2.96) se requieren 9 multiplicaciones y 6 sumas adicionales, lo que nos lleva a un total de 63 multiplicaciones y 42 sumas.

Si en lugar de eso transformamos el vector a través de las matrices, una a la vez, esto es:

$$\begin{aligned} ^{A}P &= {}^{A}_{B}R\, {}^{B}_{C}R\, {}^{C}_{D}R\, {}^{D}P \\ ^{A}P &= {}^{A}_{B}R\, {}^{B}_{C}R\, {}^{C}P \\ ^{A}P &= {}^{A}_{B}R\, {}^{B}P \\ ^{A}P &= {}^{A}P, \end{aligned} \tag{2.97}$$

entonces el cálculo total requiere solamente 27 multiplicaciones y 18 sumas; menos de la mitad de los cálculos requeridos por el otro método.

Claro que, en ciertos casos, las relaciones ${}^{B}_{A}R$, ${}^{B}_{C}R$ y ${}^{C}_{D}R$ son constantes, mientras que hay muchos ${}^{D}P_i$ que necesitan transformarse en ${}^{A}P_i$. En tal caso, es más eficiente calcular ${}^{A}_{D}R$ de una vez y luego usarlo para todas las asignaciones futuras. Vea también el ejercicio 2.16.

EJEMPLO 2.10

Proporcione un método para calcular el producto de dos matrices de rotación, ${}^A_BR\,{}^B_CR$, que utilice menos de 27 multiplicaciones y 18 sumas.

Si $\hat{L}_i$ representa a las columnas de ${}^B_CR$ y $\hat{C}_i$ representa las tres columnas del resultado, calcule

$$\begin{aligned}\hat{C}_1 &= {}^A_BR\hat{L}_i,\\ \hat{C}_2 &= {}^A_BR\hat{L}_2,\\ \hat{C}_3 &= \hat{C}_1 \times \hat{C}_2,\end{aligned} \tag{2.98}$$

que requiere 24 multiplicaciones y 15 sumas.

BIBLIOGRAFÍA

[1] B. Noble, *Applied Linear Algebra*, Prentice-Hall, Englewood Cliffs, NJ, 1969.

[2] D. Ballard y C. Brown, *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[3] O. Bottema y B. Roth, *Theoretical Kinematics*, North Holland, Amsterdam, 1979.

[4] R.P. Paul, *Robot Manipulators*, MIT Press, Cambridge, MA, 1981.

[5] I. Shames, *Engineering Mechanics*, 2a Edición, Prentice-Hall, Englewood Cliffs, NJ, 1967.

[6] Symon, *Mechanics*, 3a Edición, Addison-Wesley, Reading, MA, 1971.

[7] B. Gorla y M. Renaud, *Robots Manipulateurs*, Cepadues-Editions, Toulouse, 1984.

EJERCICIOS

2.1 [15] Un vector ${}^AP$ se gira θ grados sobre $\hat{Z}_A$ y se gira subsecuentemente ϕ grados sobre $\hat{X}_A$. Proporcione la matriz de rotación que realice estas rotaciones en el orden dado.

2.2 [15] Un vector ${}^AP$ se gira 30 grados sobre $\hat{Y}_A$ y subsecuentemente otros 45 grados sobre $\hat{X}_A$. Proporcione la matriz de rotación que realice estas rotaciones en el orden dado.

2.3 [16] Una trama $\{B\}$ se encuentra inicialmente coincidente con una trama $\{A\}$. Giramos $\{B\}$ θ grados sobre $\hat{Z}_B$ y luego giramos la trama resultante ϕ grados sobre $\hat{X}_B$. Proporcione la matriz de rotación que cambie las descripciones de los vectores de ${}^BP$ a ${}^AP$.

2.4 [16] Una trama $\{B\}$ se encuentra inicialmente coincidente con una trama $\{A\}$. Giramos $\{B\}$ 30 grados sobre $\hat{Z}_B$ y luego giramos la trama resultante 45 grados sobre $\hat{X}_B$. Proporcione la matriz de rotación que cambie la descripción de los vectores de ${}^BP$ a ${}^AP$.

2.5 [13] ${}^A_BR$ es una matriz de 3×3 con los valores propios 1, e^{+ai} y e^{-ai}; en donde $i = \sqrt{-1}$. ¿Cuál es el significado físico del vector propio de ${}^A_BR$ asociado con el valor propio 1?

2.6 [21] Derive la ecuación (2.80).

2.7 [24] Describa (o programe) un algoritmo que extraiga el ángulo equivalente y el eje de una matriz de rotación. La ecuación (2.82) es un buen comienzo, pero asegúrese que su algoritmo maneje los casos especiales $\theta = 0°$ y $\theta = 180°$.

2.8 [29] Escriba una subrutina que cambie la representación de la orientación de la forma matriz de rotación a la forma ángulo-eje equivalente. La declaración de un procedimiento estilo Pascal empezaría así:

```
Procedure MRAAE (VAR R:mat33; VAR K:vec3; VAR theta: real);
```

Escriba otra subrutina que cambie de la representación de ángulo-eje equivalente a la representación de matriz de rotación:

```
Procedure AEAMR(VAR K:vec3; VAR theta: real; VAR R:mat33);
```

Escriba las rutinas en C si lo prefiere.

Ejecute estos procedimientos en varios casos de datos de prueba extensivamente y verifique que obtenga lo que introdujo. Incluya algunos de los casos difíciles.

2.9 [27] Haga el ejercicio 2.8 para ángulos de balanceo, inclinación y orientación sobre ejes fijos.

2.10 [27] Haga el ejercicio 2.8 para ángulos de EulerZ-Y-Z.

2.11 [10] ¿Bajo qué condición se conmutan dos matrices de rotación que representan rotaciones finitas? No se requiere una prueba.

2.12 [14] Un vector de velocidad se da de la siguiente forma:

$$^{B}V = \begin{bmatrix} 10.0 \\ 20.0 \\ 30.0 \end{bmatrix}.$$

Dada

$$^{A}_{B}T = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 11.0 \\ 0.500 & 0.866 & 0.000 & -3.0 \\ 0.000 & 0.000 & 1.000 & 9.0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

calcule ^{A}V.

2.13 [21] Las siguientes definiciones de trama se dan como conocidas:

$$^{U}_{A}T = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 11.0 \\ 0.500 & 0.866 & 0.000 & -1.0 \\ 0.000 & 0.000 & 1.000 & 8.0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$^{B}_{A}T = \begin{bmatrix} 1.000 & 0.000 & 0.000 & 0.0 \\ 0.000 & 0.866 & -0.500 & 10.0 \\ 0.000 & 0.500 & 0.866 & -20.0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$^{C}_{U}T = \begin{bmatrix} 0.866 & -0.500 & 0.000 & -3.0 \\ 0.433 & 0.750 & -0.500 & -3.0 \\ 0.250 & 0.433 & 0.866 & 3.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Dibuje un diagrama de tramas (como el de la figura 2.15) para mostrar su arreglo cualitativamente y resuelva para $^{B}_{C}T$.

2.14 [31] Desarrolle una fórmula general para obtener $^{A}_{B}T$ en donde, empezando por la coincidencia inicial, $\{B\}$ se gire θ grados sobre $\hat{K}$, cuando $\hat{K}$ pase por el punto ^{A}P (no a través del origen de $\{A\}$).

2.15 [34] $\{A\}$ y $\{B\}$ son tramas que difieren sólo en su orientación. $\{B\}$ se obtiene de la siguiente manera: empezando en forma coincidente con $\{A\}$, $\{B\}$ se gira θ radianes sobre el vector unitario $\hat{K}$, esto es,

$$^{A}_{B}R = {}^{A}_{B}R_{K}(\theta).$$

Muestre que

$$ {}^{A}_{B}R = e^{k\theta}, $$

en donde

$$ K = \begin{bmatrix} 0 & -k_x & ky \\ k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{bmatrix}. $$

2.16 [22] Un vector debe asignarse a través de tres matrices de rotación:

$$ {}^{A}P = {}^{A}_{B}R\, {}^{B}_{C}R\, {}^{C}_{D}R\, {}^{D}P. $$

Una opción es multiplicar primero las tres matrices de rotación juntas para formar ${}^{A}_{D}R$ en la expresión

$$ {}^{A}P = {}^{A}_{D}R\, {}^{D}P. $$

Otra opción es transformar el vector a través de las matrices, una a la vez; esto es,

$$ {}^{A}P = {}^{A}_{B}R\, {}^{B}_{C}R\, {}^{C}_{D}R\, {}^{D}P, $$

$$ {}^{A}P = {}^{A}_{B}R\, {}^{B}_{C}R\, {}^{C}P, $$

$$ {}^{A}P = {}^{A}_{B}R\, {}^{B}P, $$

$$ {}^{A}P = {}^{A}P. $$

Si ${}^{D}P$ está cambiando a 100 Hz, tendríamos que recalcular ${}^{A}P$ a la misma proporción. No obstante, las tres matrices de rotación también están cambiando, según lo reportado por un sistema de visión que nos da nuevos valores para ${}^{A}_{B}R$, ${}^{B}_{C}R$ y ${}^{C}_{D}R$ a 30 Hz. ¿Cuál es la mejor manera de organizar el cómputo para minimizar el esfuerzo de cálculo (multiplicaciones y sumas)?

2.17 [16] Otro conjunto familiar de tres coordenadas que puede utilizarse para describir un punto en el espacio son las coordenadas cilíndricas. Las tres coordenadas se definen como se muestra en la figura 2.23. La coordenada θ proporciona una dirección en el plano xy sobre el cual se realiza la traslación radialmente con base en una cantidad r. Finalmente se da z para especificar la altura sobre el plano xy. Calcule las coordenadas cartesianas del punto ${}^{A}P$ en términos de las coordenadas cilíndricas θ, r y z.

2.18 [18] Otro conjunto de tres coordenadas que puede usarse para describir un punto en el espacio son las coordenadas esféricas. Las tres coordenadas se definen según se muestra en la figura 2.24. Se puede considerar que los ángulos α y β describen el acimut y la elevación de un rayo que se proyecta en el espacio. La tercera coordenada (r) es la distancia radial sobre ese rayo hasta el punto que se está describiendo. Calcule las coordenadas cartesianas del punto ${}^{A}P$ en términos de las coordenadas esféricas α, β y r.

2.19 [24] Un objeto se gira una cantidad de ϕ grados sobre su eje $\hat{X}$ y después se gira una cantidad de Ψ sobre su nuevo eje $\hat{Y}$. De nuestro estudio de los ángulos de Euler sabemos que la orientación resultante se da mediante

$$ R_x(\phi)R_y(\psi), $$

mientras que, si las dos rotaciones hubieran ocurrido sobre ejes de la trama de referencia fija, el resultado hubiera sido

$$ R_y(\psi)R_x(\phi). $$

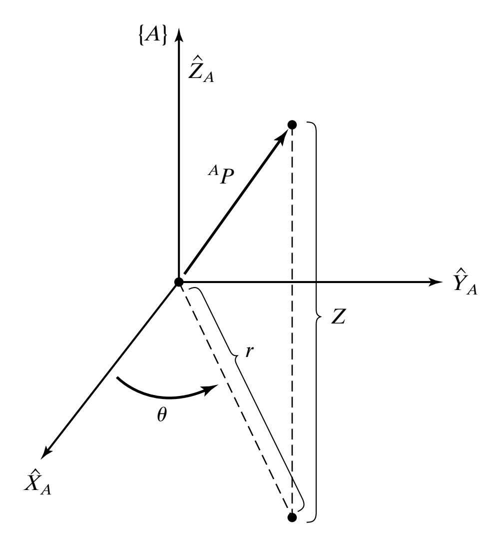


FIGURA 2.23: Coordenadas cilíndricas.

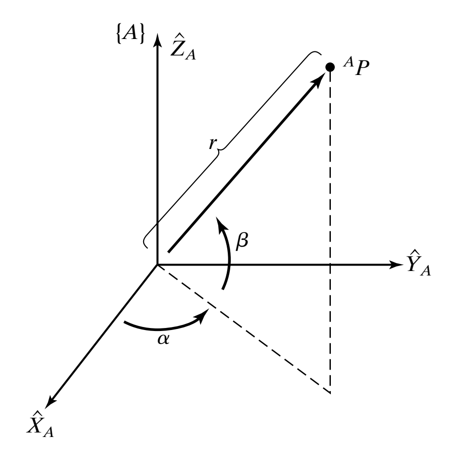


FIGURA 2.24: Coordenadas esféricas.

Parece que el orden de multiplicación depende de si las rotaciones se describen en forma relativa a los ejes fijos o a los de la trama que se está moviendo. No obstante, es más apropiado tener en cuenta que en caso de especificar una rotación sobre un eje de la trama que se está moviendo, estamos especificando una rotación en el sistema fijo que (en este ejemplo) está dado por:

$$R_x(\phi)R_y(\psi)R_x^{-1}(\phi).$$

Esta *transformada de similitud* [1], que multiplica a la $R_x(\phi)$ de la izquierda, se reduce a la expresión resultante en la cual *se ve como si* se hubiera invertido el orden de multiplicación de las matrices. Tomando este punto de vista, proporcione una derivación

para la forma de la matriz de rotación que sea equivalente al conjunto de ángulos de Euler Z-Y-X (α, β, γ). El resultado se da mediante la ecuación (2.72).

2.20 [20] Imagine que se gira un vector Q sobre un vector $\hat{K}$ una cantidad de θ grados para formar un nuevo vector, Q'; esto es,

$$Q' = R_K(\theta)Q.$$

Use la ecuación (2.80) para derivar la **fórmula de Rodrigues**:

$$Q' = Q\cos\theta + \operatorname{sen}\theta(\hat{K} \times Q) + (1 - \cos\theta)(\hat{K} \cdot \hat{Q})\hat{K}.$$

2.21 [15] Para rotaciones que sean suficientemente pequeñas como para que las aproximaciones sen $\theta = \theta$, cos $\theta = 1$, y $\theta^2 = 0$ sean siempre ciertas, derive la matriz de rotación equivalente a una rotación de θ sobre un eje general, $\hat{K}$. Empiece con la ecuación (2.80) para su derivación.

2.22 [20] Usando el resultado del ejercicio 2.21, muestre que dos rotaciones infinitesimales conmutan (es decir, el orden en el que se realizan las rotaciones no es importante).

2.23 [25] Proporcione un algoritmo para construir la definición de una trama ${}^U_A T$ a partir de tres puntos ${}^U P_1$, ${}^U P_2$ y ${}^U P_3$, en donde se conoce lo siguiente sobre estos puntos:

1. ${}^U P_1$ se encuentra en el origen de $\{A\}$.
2. ${}^U P_2$ se encuentra en alguna parte del eje $\hat{X}$ positivo de $\{A\}$.
3. ${}^U P_3$ se encuentra cerca del eje $\hat{Y}$ positivo, en el plano XY de $\{A\}$.

2.24 [45] Demuestre la fórmula de Cayley para matrices ortonormales propias.

2.25 [30] Muestre que los valores propios de una matriz de rotación son 1, e^{ai} y e^{-ai}, en donde $i = -1$.

2.26 [33] Demuestre que cualquier conjunto de ángulos de Euler es suficiente para expresar todas las posibles matrices de rotación.

2.27 [15] Tomando como referencia la figura 2.25, proporcione el valor de ${}^A_B T$.

2.28 [15] Tomando como referencia la figura 2.25, proporcione el valor de ${}^A_C T$.

2.29 [15] Tomando como referencia la figura 2.25, proporcione el valor de ${}^B_C T$.

2.30 [15] Tomando como referencia la figura 2.25, proporcione el valor de ${}^C_A T$.

2.31 [15] Tomando como referencia la figura 2.26, proporcione el valor de ${}^A_B T$.

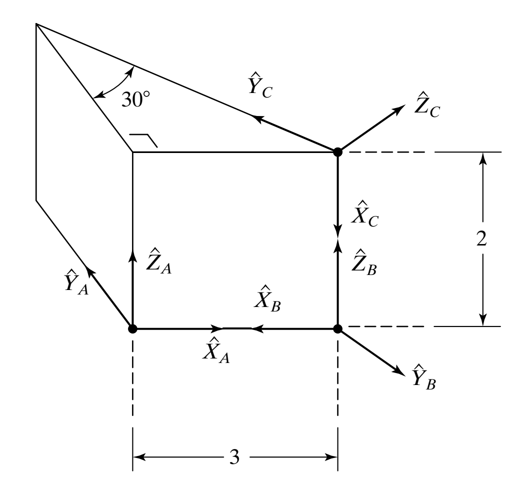


FIGURA 2.25: Tramas en las esquinas de una cuña.

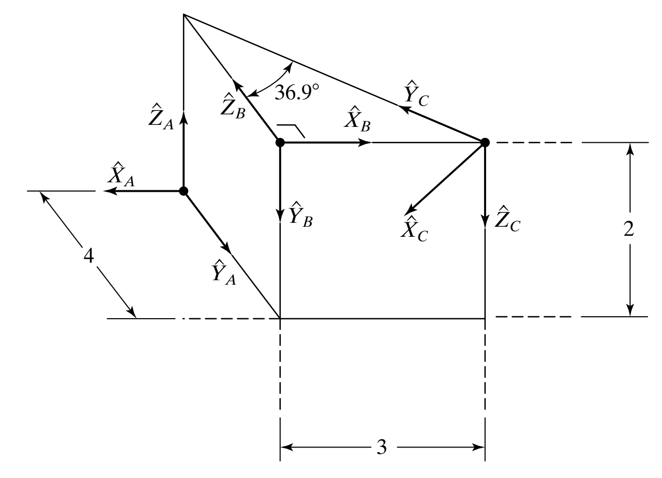


FIGURA 2.26: Tramas en las esquinas de una cuña.

2.32 [15] Tomando como referencia la figura 2.26, proporcione el valor de ${}^{A}_{C}T$.
2.33 [15] Tomando como referencia la figura 2.26, proporcione el valor de ${}^{B}_{C}T$.
2.34 [15] Tomando como referencia la figura 2.26, proporcione el valor de ${}^{C}_{A}T$.
2.35 [20] Demuestre que el determinante de cualquier matriz de rotación es siempre igual a 1.
2.36 [36] Un cuerpo rígido que se desplaza en un plano (por ejemplo, en espacio 2D) tiene tres grados de libertad. Un cuerpo rígido que se desplaza en espacio 3D tiene seis grados de libertad. Muestre que un cuerpo en espacio N-dimensional tiene $\frac{1}{2}(N^2 + N)$ grados de libertad.
2.37 [15] Dada

$$ {}^{A}_{B}T = \begin{bmatrix} 0.25 & 0.43 & 0.86 & 5.0 \\ 0.87 & -0.50 & 0.00 & -4.0 \\ 0.43 & 0.75 & -0.50 & 3.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

¿cuál es el elemento (2, 4) de ${}^{B}_{A}T$?
2.38 [25] Imagine dos vectores unitarios, v_1 y v_2, incrustados en un cuerpo rígido. Tenga en cuenta que, sin importar cómo se gire el cuerpo, el ángulo geométrico entre estos dos vectores se preserva (es decir, la rotación de un cuerpo rígido es una operación de “preservación de ángulo”). Use este hecho para proporcionar una prueba concisa (de cuatro o cinco líneas) de que la inversa de una matriz de rotación debe ser igual a su transpuesta, y de que una matriz de rotación es ortonormal.
2.39 [37] Proporcione un algoritmo (tal vez en forma de programa en C) que calcule el cuaternión unitario que corresponda a una matriz de rotación dada. Utilice (2.91) como punto de inicio.
2.40 [33] Proporcione un algoritmo (tal vez en forma de un programa en C) que calcule los ángulos de Euler Z-X-Z que correspondan a una matriz de rotación dada. Vea el apéndice B.
2.41 [33] Proporcione un algoritmo (tal vez en forma de un programa en C) que calcule los ángulos fijos X-Y-X que correspondan a una matriz de rotación dada. Vea el apéndice B.

EJERCICIO DE PROGRAMACIÓN (PARTE 2)

1. Si su biblioteca de funciones no incluye una subrutina de función Atan2, escriba una.
2. Para hacer una interfaz de usuario amigable, deseamos describir la orientación en el mundo planar mediante un solo ángulo θ, en vez de utilizar una matriz de rotación de

2 × 2. El usuario siempre se comunicará en términos del ángulo θ, pero internamente necesitaremos la forma de la matriz de rotación. Para la parte correspondiente al vector de posición de una trama, el usuario especificará un valor para x y uno para y. Por lo tanto, queremos permitir al usuario que especifique a la *trama* como una tríada: (x, y, θ). Internamente deseamos usar un vector de posición de 2 × 1 y una matriz de rotación de 2 × 2, por lo que necesitamos rutinas de conversión. Escriba una rutina cuya definición en Pascal empiece así:

```
Procedure UAI (VAR formau: vec3; VAR formai: trama);
```

en donde "UAI " significa "formato de Usuario A formato Interno". El primer argumento es la tríada (x, y, θ) y el segundo argumento es de tipo "trama", el cual consiste en un vector de posición (2 × 1) y una matriz de rotación (2 × 2). Si lo desea, puede representar la trama con una transformada homogénea (3 × 3) en la que la tercera fila sea [0 0 1]. También será necesaria la rutina inversa:

```
Procedure IAU (VAR formai: trama; VAR formau: vec3);
```

3. Escriba una subrutina para multiplicar dos transformadas. Use el siguiente encabezado de procedimiento:

```
Procedure  TMULT (VAR brela, crelb, crela: trama);
```

Los primeros dos argumentos son entradas y el tercero es salida. Observe que los nombres de los argumentos documentan lo que hace el programa (brela $= {}^{A}_{B}T$).

4. Escriba una subrutina para invertir una transformada. Use el siguiente encabezado de procedimiento:

```
Procedure TINVERT (VAR brela, arelb: trama);
```

El primer argumento es la entrada, el segundo la salida. Observe que los nombres de los argumentos documentan lo que hace el programa (brela $= {}^{A}_{B}T$).

5. Las siguientes definiciones de tramas se dan como conocidas:

$$ {}^{U}_{A}T = [x\ y\ \theta] = [11.0\ \ -1.0\ 30.0], $$

$$ {}^{B}_{A}T = [x\ y\ \theta] = [0.0\ 7.0\ 45.0], $$

$$ {}^{C}_{U}T = [x\ y\ \theta] = [-3.0\ \ -3.0\ \ -30.0]. $$

Estas tramas se introducen en la representación del usuario $[x, y, \theta]$ (en donde θ se da en grados). Dibuje un diagrama de tramas (como el de la figura 2.15, sólo que en 2D) que muestre cualitativamente su arreglo. Escriba un programa que llame a TMULT y TINVERT (definidas en los ejercicios de programación 3 y 4) todas las veces necesarias para resolver para ${}^{B}_{C}T$. Después imprima ${}^{B}_{C}T$ tanto en representación interna como del usuario.

EJERCICIO MATLAB 2A

a) Utilizando la convención de ángulos de Euler Z-Y-X $(\alpha - \beta - \gamma)$, escriba un programa MATLAB para calcular la matriz de rotación ${}^{A}_{B}R$ cuando el usuario introduzca los ángulos de Euler $\alpha - \beta - \gamma$. Haga la prueba en dos ejemplos:

i) $\alpha = 10°$, $\beta = 20°$, $\gamma = 30°$.
iii) $\alpha = 30°$, $\beta = 90°$, $\gamma = -55°$.

Para el caso (**i**) demuestre las seis restricciones para matrices de rotación ortonormales unitarias (hay nueve números en una matriz de 3 × 3, pero sólo tres son independientes). Además, demuestre la *hermosa* propiedad, ${}^{B}_{A}R = {}^{A}_{B}R^{-1} = {}^{A}_{B}R^{T}$, para el caso **i**.

b) Escriba un programa MATLAB para calcular los ángulos $\alpha - \beta - \gamma$ de Euler cuando el usuario introduzca la matriz de rotación ${}^{A}_{B}R$ (el problema inverso). Calcule las dos soluciones posibles. Demuestre esta solución inversa para los dos casos de la parte (a). Use una comprobación circular para verificar sus resultados (es decir, introduzca los ángulos de Euler en el código *a* de la parte (a); tome la matriz de rotación ${}^{A}_{B}R$ resultante y utilícela como entrada para el código *b*; obtendrá dos conjuntos de respuestas: una debe ser la entrada original del usuario y la otra puede verificarse una vez más utilizando el código en la parte (a).

c) Para una rotación simple de β sobre el eje Y solamente, para $\beta = 20°$ y ${}^{B}P = \{1\ 0\ 1\}^T$, calcule ${}^{A}P$; demuestre con un bosquejo que sus resultados son correctos.

d) Compruebe todos los resultados mediante la Caja de Herramientas de Robótica MATLAB de Corke. Pruebe las funciones *rpy2tr*(), *rotx*(), *roty*() y *rotz*().

EJERCICIO MATLAB 2B

a) Escriba un programa MATLAB para calcular la matriz de transformación homogénea ${}^{A}_{B}T$ cuando el usuario introduzca los ángulos Z-Y-X de Euler $\alpha - \beta - \gamma$ y el vector de posición ${}^{A}P_B$. Haga la prueba en dos ejemplos:

i) $\alpha = 10°$, $\beta = 20°$, $\gamma = 30°$ y ${}^{A}P_B = \{1\ 2\ 3\}^T$.

ii) Para $\beta = 20°$ ($\alpha = \gamma = 0°$), ${}^{A}P_B = \{3\ 0\ 1\}^T$.

b) Para $\beta = 20°$ ($\alpha = \gamma = 0°$), ${}^{A}P_B = \{3\ 0\ 1\}^T$ y ${}^{B}P = \{1\ 0\ 1\}^T$, utilice MATLAB para calcular ${}^{A}P$; demuestre con un bosquejo que sus resultados son correctos. Además, utilizando los mismos números demuestre las tres interpretaciones de la matriz de transformación homogénea; la asignación (b) es la segunda interpretación, asignación de transformadas.

c) Escriba un programa MATLAB para calcular la matriz de transformación homogénea inversa ${}^{A}_{B}T^{-1} = {}^{B}_{A}T$, utilizando la fórmula simbólica. Compare su resultado con una función MATLAB numérica (por ejemplo, *inv*). Demuestre que ambos métodos producen resultados correctos (es decir, que ${}^{A}_{B}T\,{}^{A}_{B}T^{-1} = {}^{A}_{B}T^{-1}\,{}^{A}_{B}T = I_4$). Demuéstrelo para los ejemplos (i) y (iii) del inciso (a) anterior.

d) Defina ${}^{A}_{B}T$ para que sea el resultado de (a)(i) y que ${}^{B}_{C}T$ sea el resultado de (a)(ii).

i) Calcule ${}^{A}_{C}T$ y muestre la relación mediante un gráfico de transformadas. Haga lo mismo para ${}^{C}_{A}T$.

ii) Dadas ${}^{A}_{C}T$ y ${}^{B}_{C}T$ de (d)(i), suponga que no conoce ${}^{A}_{B}T$, calcúlela y compare su resultado con la respuesta que conoce.

iii) Dadas ${}^{A}_{C}T$ y ${}^{A}_{B}T$ de (d)(i), suponga que no conoce ${}^{B}_{C}T$, calcúlela y compare su resultado con la respuesta que conoce.

e) Compruebe todos los resultados mediante la Caja de Herramientas de Robótica MATLAB de Corke. Pruebe las funciones *rpy2tr*() y *transl*().

C A P Í T U L O 3

Cinemática de manipuladores

3.1 INTRODUCCIÓN

La cinemática es la ciencia del movimiento que trata el tema sin considerar las fuerzas que lo ocasionan. Dentro de esta ciencia se estudian la posición, la velocidad, la aceleración y todas las demás derivadas de alto orden de las variables de posición (con respecto al tiempo o a cualquier otra variable). En consecuencia, el estudio de la cinemática de manipuladores se refiere a todas las propiedades geométricas y basadas en el tiempo del movimiento. Las relaciones entre estos movimientos y las fuerzas y momentos de torsión que los ocasionan constituyen el problema de la dinámica, que es el tema del capítulo 6.

En este capítulo consideramos la posición y la orientación de los vínculos de los manipuladores en situaciones estáticas. En los capítulos 5 y 6 consideraremos la cinemática cuando hay velocidades y aceleraciones implicadas.

Para entender la compleja geometría de un manipulador, agregaremos tramas a las diversas partes del mecanismo y luego describiremos las relaciones entre estas tramas. El estudio de la cinemática de manipuladores se relaciona, entre otras cosas, con la manera en que cambian las ubicaciones de estas tramas a medida que se articula el mecanismo. El tema central de este capítulo es un método para calcular la posición y la orientación del efector final del manipulador relativo a la base del mismo, como una función de las variables de las articulaciones.

3.2 DESCRIPCIÓN DE VÍNCULOS

Un manipulador puede considerarse como un conjunto de cuerpos conectados en una cadena mediante articulaciones. Estos cuerpos se llaman vínculos o segmentos. Las articulaciones forman una conexión entre un par adyacente de vínculos. El término **par menor** se utiliza para describir la conexión entre un par de cuerpos, cuando el movimiento

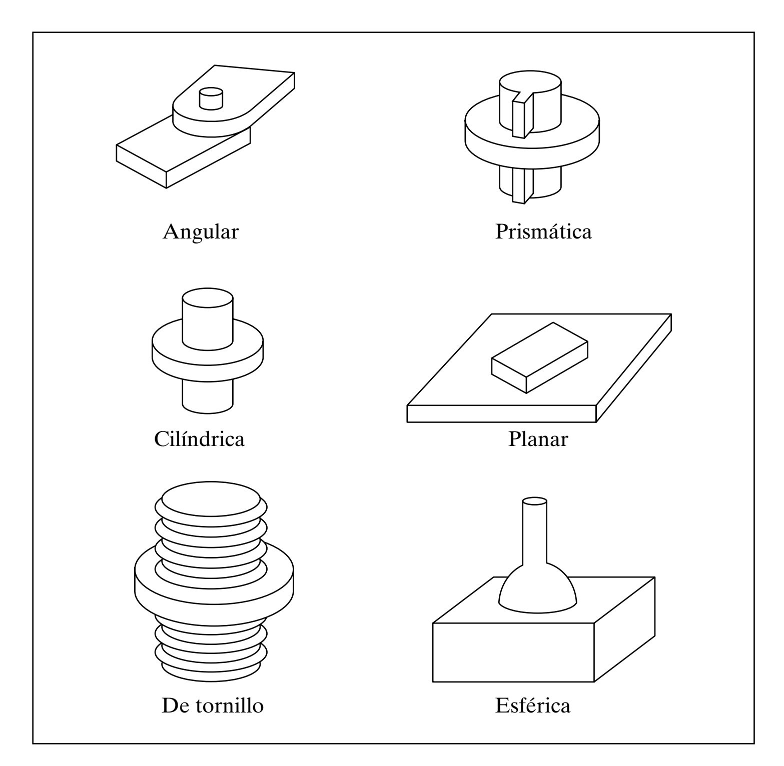


FIGURA 3.1: Las seis posibles articulaciones de par menor.

relativo se caracteriza por dos superficies que se deslizan una sobre otra. La figura 3.1 muestra las seis posibles articulaciones de par menor.

Las consideraciones de diseño mecánico recomiendan que los manipuladores se construyan generalmente de articulaciones que exhiban sólo un grado de libertad. La mayoría de los manipuladores tienen **articulaciones angulares** o articulaciones deslizantes llamadas **articulaciones prismáticas**. En el extraño caso de que un mecanismo esté construido con una articulación que tenga n grados de libertad, puede modelarse como n articulaciones de un grado de libertad, conectadas con $n - 1$ vínculos de longitud cero. Por lo tanto, y sin perder la generalidad, consideraremos sólo manipuladores que tengan articulaciones con un solo grado de libertad.

Los vínculos se numeran empezando desde la base inmóvil del brazo, a la cual podríamos llamar vínculo 0. El primer cuerpo móvil es el vínculo 1 y así, sucesivamente, hasta llegar al extremo libre del brazo, el cual es el vínculo n. Para poder posicionar un efector final en espacio 3D de forma general se requiere un mínimo de seis articulaciones.[1] Los manipuladores comunes tienen cinco o seis articulaciones. Algunos robots no son realmente tan simples como una cadena cinemática abierta; tienen una configuración de paralelogramo u otras estructuras cinemáticas cerradas. Consideraremos uno de esos manipuladores más adelante en este capítulo.

Un solo vínculo de un robot común tiene muchos atributos que el ingeniero mecánico tuvo que considerar durante su diseño: el tipo de material empleado, la fuerza y

[1] Esto tiene buen sentido intuitivo, ya que la descripción de un objeto en el espacio requiere seis parámetros: tres para la posición y tres para la orientación.

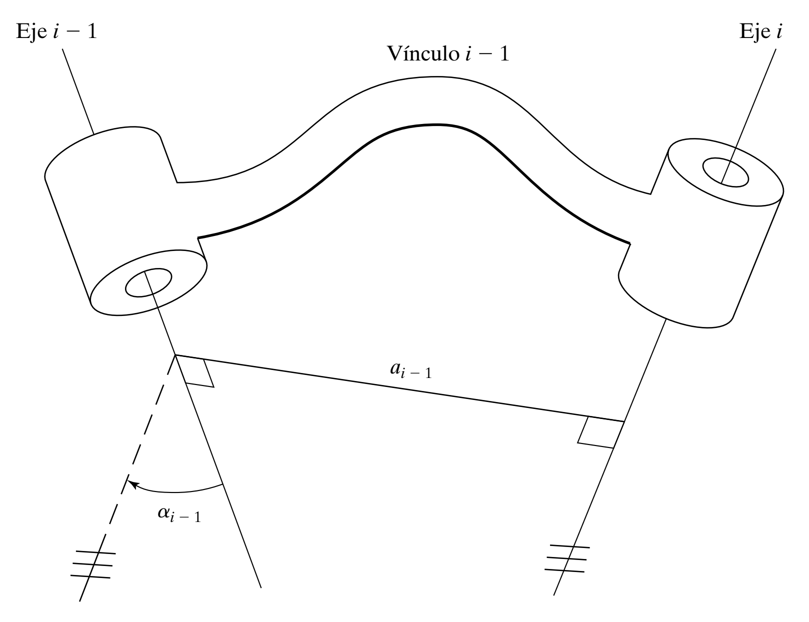


FIGURA 3.2: La función cinemática de un vínculo es mantener una relación fija entre los dos ejes de articulación que soporta. Esta relación puede describirse con dos parámetros: la longitud del vínculo, a, y su ángulo de torsión, α.

rigidez del vínculo, la ubicación y el tipo de los cojinetes de articulación, la forma externa, el peso y la inercia, y otros más. Sin embargo, cuando se desean obtener las ecuaciones cinemáticas del mecanismo, *un vínculo se considera solamente como un cuerpo rígido que define la relación entre dos ejes de articulaciones adyacentes de un manipulador*. Los ejes de articulación se definen mediante líneas en el espacio. El eje de articulación i se define mediante una línea en el espacio, o la dirección de un vector, sobre el cual gira el vínculo i con respecto al vínculo $i-1$. Resulta que, para los fines cinemáticos, un vínculo puede especificarse con dos números, los cuales definen la ubicación relativa de los dos ejes en el espacio.

Para cualesquiera dos ejes en el espacio 3D, existe una medida bien definida de distancia entre ellos. Esta distancia se mide a lo largo de una línea que es mutuamente perpendicular a ambos ejes. Esta línea mutuamente perpendicular siempre existe; es única excepto cuando ambos ejes son paralelos, en cuyo caso hay muchas líneas mutuamente perpendiculares de igual longitud. La figura 3.2 muestra el vínculo $i-1$ y la línea mutuamente perpendicular sobre la cual se mide la **longitud de vínculo** a_{i-1}. Otra manera de visualizar el parámetro del vínculo a_{i-1} es imaginar un cilindro en expansión cuyo eje es el eje de articulación $i-1$; cuando apenas toca el eje de articulación i, el radio del cilindro es igual a a_{i-1}.

El segundo parámetro necesario para definir la ubicación relativa de los dos ejes se llama la **torsión del vínculo**. Si imaginamos un plano cuya normal sea la línea mutuamente perpendicular que se acaba de construir, podemos proyectar los ejes $i-1$ e i en este plano y medir el ángulo entre ellos. Este ángulo se mide desde el eje $i-1$ hasta el eje i en sentido de la mano derecha, sobre a_{i-1}.[2] Utilizaremos esta definición de la torsión del vínculo $i-1$, α_{i-1}. En la figura 3.2, α_{i-1} se indica como el ángulo entre el eje

[2] En este caso, a a_{i-1} se le asigna la dirección que apunta desde el eje $i-1$ hasta el eje i.

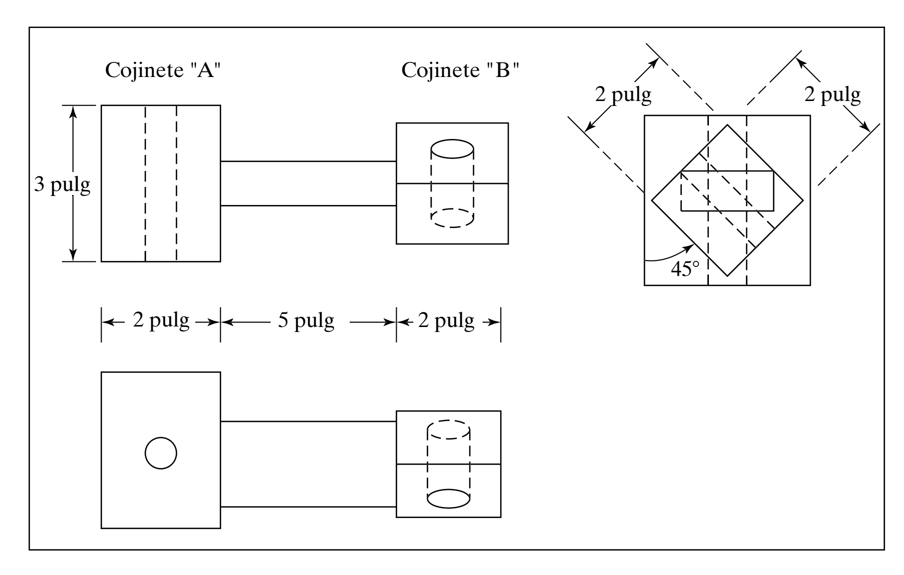


FIGURA 3.3: Un vínculo simple que soporta dos ejes angulares.

$i - 1$ y el eje i (las líneas con tres marcas de referencia son paralelas). En el caso de ejes perpendiculares, la torsión se mide en el plano que contiene ambos ejes pero se pierde el sentido de α_{i-1}. En este caso especial tenemos la libertad de asignar el signo de α_{i-1} arbitrariamente.

Estos dos parámetros, longitud y torsión, según se definieron anteriormente, pueden usarse para definir también la relación entre dos líneas cualesquiera (en este caso ejes) en el espacio.

EJEMPLO 3.1

La figura 3.3 muestra los dibujos mecánicos de un vínculo de robot. Si este vínculo se utiliza en un robot en donde se use el cojinete "A" para la articulación de menor numeración, proporcione la longitud y torsión de este vínculo. Suponga que los hoyos están centrados en cada cojinete.

Por inspección, la perpendicular común está justo debajo de la parte media de la barra de metal que conecta los cojinetes, por lo que la longitud del vínculo es de 7 pulgadas. La vista lateral muestra una proyección de los cojinetes en el plano cuya normal es la perpendicular mutua. La torsión del vínculo se mide en el sentido de la mano derecha sobre la perpendicular común, desde el eje $i - 1$ hasta el eje i, por lo que en este ejemplo puede verse claramente que es de $+45$ grados.

3.3 DESCRIPCIÓN DE LA CONEXIÓN DE VÍNCULOS

El problema de conectar los vínculos de un robot entre sí también está lleno de preguntas que el ingeniero mecánico debe resolver. Aquí se incluyen la solidez de la articulación, su lubricación y el montaje del cojinete y el engranaje. No obstante, para la investigación de la cinemática sólo necesitamos preocuparnos por dos cantidades, las cuales especifican completamente la manera en que se conectan los vínculos entre sí.

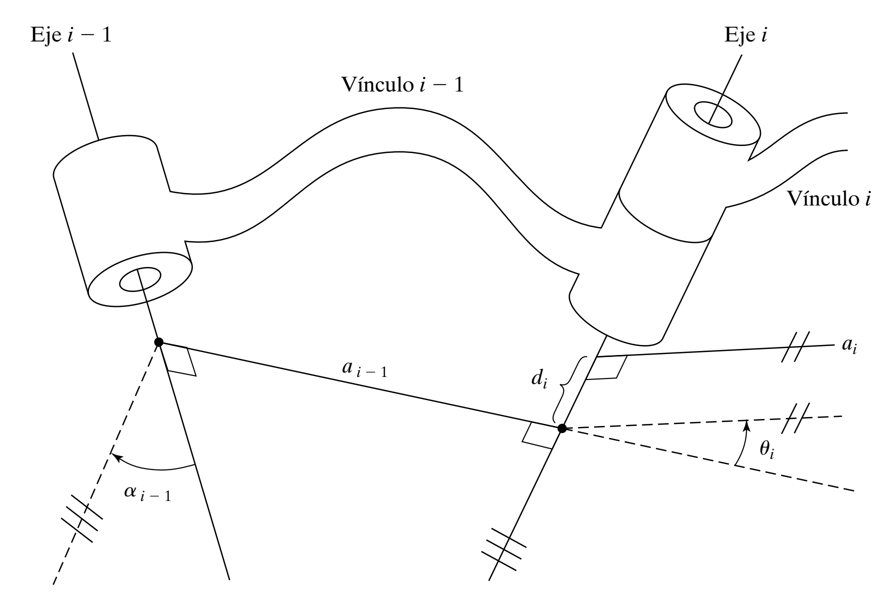


FIGURA 3.4: El desplazamiento del vínculo d y el ángulo de articulación θ, son dos parámetros que pueden usarse para describir la naturaleza de la conexión entre vínculos adyacentes.

Vínculos intermedios en la cadena

Los vínculos adyacentes tienen un eje de articulación común entre ellos. Uno de los parámetros de interconexión tiene que ver con la distancia a lo largo de este eje común, desde un vínculo hasta el siguiente. Este parámetro se llama **desplazamiento del vínculo**. El desplazamiento en el eje de articulación i se llama d_i. El segundo parámetro describe la cantidad de rotación sobre este eje común entre un vínculo y su vínculo adyacente. Este parámetro se llama **ángulo de articulación**, θ_i.

La figura 3.4 muestra la interconexión del vínculo i–1 y el vínculo i. Recuerde que a_{i-1} es la perpendicular mutua entre los dos ejes del vínculo i–1. De igual forma, a_i es la perpendicular mutua definida para el vínculo i. El primer parámetro de interconexión es el desplazamiento del vínculo d_i, que es la distancia con signo que se mide a lo largo del eje de articulación i, desde el punto en el que a_{i-1} se interseca con el eje hasta el punto en el que a_i se interseca con el eje. El desplazamiento d_i, se indica en la figura 3.4. El desplazamiento del vínculo d_i es variable si la articulación i es prismática. El segundo parámetro de interconexión es el ángulo que se forma entre una extensión de a_{i-1} y a_i, que se mide sobre el eje de la articulación i. Esto se indica en la figura 3.4, en donde las líneas con las marcas de referencia dobles son paralelas. A este parámetro se le llama θ_i y es variable para una articulación angular.

Primer y último vínculos en la cadena

La longitud de vínculo a_i y la torsión del vínculo α_i dependen de los ejes de articulación i e $i+1$. Por ende, desde a_1 hasta a_{n-1} y desde α_1 hasta α_{n-1} se definen según lo que vimos en esta sección. En los extremos de la cadena nuestra convención será asignar cero a estas cantidades. Esto es, $a_0 = a_n = 0.0$ y $\alpha_0 = \alpha_n = 0.0$.[3] El desplazamiento del

[3] De hecho, a_n y α_n no necesitan definirse.

vínculo d_i y el ángulo de articulación θ_i están bien definidos para las articulaciones 2 hasta $n - 1$, de acuerdo con las convenciones que vimos en esta sección. Si la articulación 1 es angular, la posición cero para θ_1 puede elegirse arbitrariamente; $d_1 = 0.0$ será nuestra convención. De manera similar, si la articulación 1 es prismática, la posición cero de d_1 puede elegirse arbitrariamente; $\theta_1 = 0$ será nuestra convención. Lo mismo se aplica exactamente igual a la articulación n.

Estas convenciones se han seleccionado de manera que, en el caso en que una cantidad pudiera asignarse arbitrariamente, se asigna un valor de cero para que los cálculos posteriores sean lo más simples que se pueda.

Parámetros de vínculo

Cualquier robot puede describirse en forma cinemática proporcionando los valores de cuatro cantidades para cada vínculo. Dos describen el vínculo en sí, y los otros dos describen la conexión del vínculo con un vínculo adyacente. En el caso de una articulación angular, θ_i se llama **variable de articulación** y las otras tres cantidades son **parámetros de vínculo** fijos. Para las articulaciones prismáticas, d_i es la variable de articulación y las otras tres cantidades son parámetros de vínculo fijos. La definición de mecanismos por medio de estas cantidades es una convención que generalmente se le conoce como **notación Denavit-Hartenberg** [1].[4] Hay otros métodos disponibles para describir mecanismos, pero no los presentaremos aquí.

En este punto podríamos inspeccionar cualquier mecanismo y determinar los parámetros de Denavit-Hartenberg que lo describen. Para un robot con seis articulaciones se requerirían 18 números para describir completamente la porción fija de su cinemática. En el caso de un robot con seis articulaciones, en donde todas son angulares, los 18 números se encuentran como seis conjuntos de la forma (α_i, α_i, d_i).

EJEMPLO 3.2

Dos vínculos, según se describe en la figura 3.3, se conectan como los vínculos 1 y 2 de un robot. La articulación 2 está compuesta de un cojinete "B" del vínculo 1 y un cojinete "A" del vínculo 2, ordenados de manera que las superficies planas de los cojinetes "A" y "B" encajen entre sí. ¿Cuánto vale d_2?

El desplazamiento de vínculo d_2 es el de la articulación 2, que viene siendo la distancia medida a lo largo del eje de la articulación 2, entre la perpendicular mutua del vínculo 1 y la del vínculo 2. De los dibujos de la figura 3.3 se puede ver que es de 2.5 pulgadas.

Antes de introducir más ejemplos, definiremos una convención para adjuntar una trama a cada vínculo del manipulador.

3.4 CONVENCIÓN PARA ASIGNAR TRAMAS A VÍNCULOS

Para poder describir la ubicación de cada vínculo relativa a sus vínculos adyacentes, definimos una trama unida a cada vínculo. Las tramas de los vínculos se nombran por número, de acuerdo al vínculo al cual están unidas. Es decir, la trama $\{i\}$ se une rígidamente al vínculo i.

[4] Tenga en cuenta que muchas convenciones relacionadas tienen el nombre Denavit-Hartenberg, pero difieren en algunos detalles. Por ejemplo, la versión utilizada en este libro difiere de otros libros de robótica en la manera de numerar tramas. A diferencia de otras convenciones, en este libro la trama $\{i\}$ está unida al vínculo i y tiene su origen en el eje de articulación i.

Vínculos intermedios en la cadena

La convención que utilizaremos para ubicar tramas en los vínculos es la siguiente: El eje $\hat{Z}$ de la trama $\{i\}$, llamado $\hat{Z}_i$, es coincidente con el eje de articulación i. El origen de la trama $\{i\}$ se ubica en donde la perpendicular a_i se interseca con el eje de articulación i. $\hat{X}_i$ apunta hacia a_i en la dirección de la articulación i hacia la articulación $i + 1$.

En el caso de $a_i = 0$, $\hat{X}_i$ es normal al plano de $\hat{Z}_i$ y de $\hat{Z}_{i+1}$. Definimos α_i midiéndola en sentido de la mano derecha sobre $\hat{X}_i$, y así podemos ver que la libertad de elegir el signo de a_i en este caso corresponde a dos opciones para la dirección de $\hat{X}_i$. $\hat{Y}_i$ se forma mediante la regla de la mano derecha para completar la i-ésima trama. La figura 3.5 muestra la ubicación de las tramas $\{i\text{-}1\}$ e $\{i\}$ para un manipulador general.

Primero y último vínculos en la cadena

Adjuntamos una trama a la base del robot, o vínculo 0, a la cual llamaremos trama $\{0\}$. Esta trama no se mueve; para el problema cinemático del brazo puede considerarse como la trama de referencia. Podemos describir la posición de todas las demás tramas de vínculos en términos de esta trama.

La trama $\{0\}$ es arbitraria, por lo que siempre es más simple elegir $\hat{Z}_0$ sobre el eje 1 y ubicar la trama $\{0\}$ de manera que coincida con la trama $\{1\}$ cuando la variable de articulación 1 es cero. Usando esta convención, siempre tendremos que $a_0 = 0.0$, $\alpha_0 = 0.0$. Además, esto asegura que $d_1 = 0.0$ si la articulación 1 es angular, o que $\theta_1 = 0.0$ si la articulación 1 es prismática.

Para la articulación n angular, la dirección de $\hat{X}_N$ se selecciona de manera que siempre se alinee con $\hat{X}_{N-1}$ cuando $\theta_n = 0.0$, y el origen de la trama $\{N\}$ se selecciona de manera que $d_n = 0.0$. Para la articulación n prismática, la dirección de $\hat{X}_N$ se selecciona de manera que $\theta_n = 0.0$, y el origen de la trama $\{N\}$ se selecciona en la intersección de $\hat{X}_{N-1}$ y el eje de articulación n cuando $d_n = 0.0$.

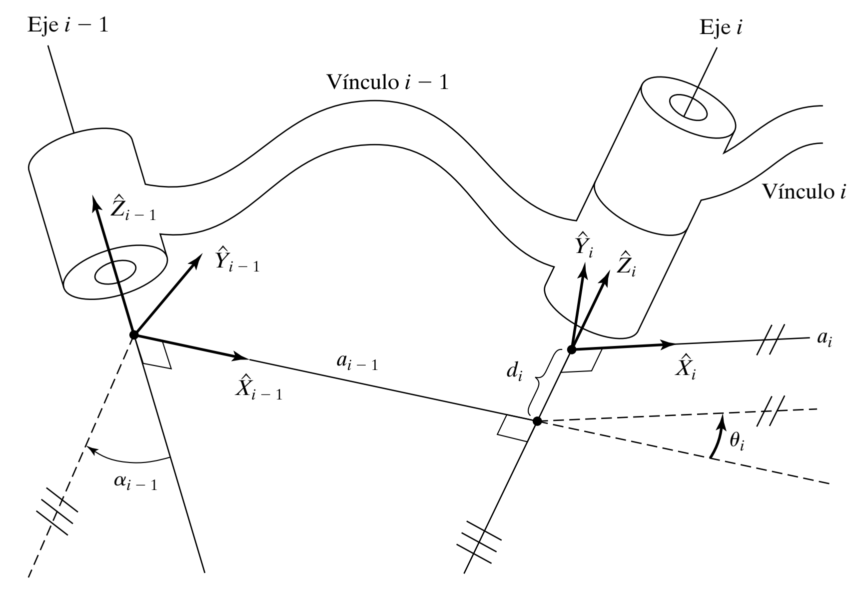


FIGURA 3.5: Las tramas de los vínculos se unen de manera que la trama $\{i\}$ se una rígidamente al vínculo i.

Resumen de los parámetros de vínculo en términos de las tramas de los vínculos

Si las tramas de los vínculos son asignadas de acuerdo con nuestra convención, son válidas las siguientes definiciones:

a_i = *la distancia de* $\hat{Z}_i$ *a* $\hat{Z}_{i+1}$ *medida sobre* $\hat{X}_i$;

α_i = *el ángulo de* $\hat{Z}_i$ *a* $\hat{Z}_{i+1}$ *medido sobre* $\hat{X}_i$;

d_i = *la distancia de* $\hat{X}_{i-1}$ *a* $\hat{X}_i$ *medida sobre* $\hat{Z}_i$; y

θ_i = *el ángulo de* $\hat{X}_{i-1}$ *a* $\hat{X}_i$ *medido sobre* $\hat{Z}_i$.

Generalmente elegimos $a_i > 0$ ya que corresponde a una distancia; no obstante α_i, d_i y θ_i son cantidades con signo.

Hay que tener en cuenta una observación final sobre la unicidad. La convención descrita anteriormente no representa la única manera de asignar tramas a los vínculos. Primero que nada, cuando alineamos el eje $\hat{Z}_i$ con el eje de articulación i hay dos opciones de dirección en las cuales puede apuntar $\hat{Z}_i$. Lo que es más, en el caso de ejes de articulación que se intersecan (es decir, que $a_i = 0$) hay dos opciones para la dirección de $\hat{X}_i$, que corresponden a la opción de signos para la normal en el plano que contenga $\hat{Z}_i$ y $\hat{Z}_{i+1}$. Cuando los ejes i e $i + 1$ son paralelos, la opción de ubicación del origen para $\{i\}$ es arbitraria (aunque generalmente se selecciona para poder hacer que d_i sea cero). Además, cuando hay articulaciones prismáticas presentes, hay mucha libertad en la asignación de tramas. (Vea también el ejemplo 3.5).

Resumen del procedimiento para asignar tramas a vínculos

A continuación se presenta un resumen del procedimiento a seguir, cuando nos encontramos con un nuevo mecanismo, para asignar correctamente las tramas de los vínculos:

1. Identifique los ejes de articulación e imagine (o dibuje) líneas infinitas sobre ellos. Para los pasos del 2 al 5, considere dos de estas líneas adyacentes (en los ejes i e $i + 1$).
2. Identifique la perpendicular común entre ellos, o el punto de intersección. En el punto de intersección, o en el punto en el que la perpendicular común se encuentra con el i-ésimo eje, asigne el origen de la trama asociada al vínculo.
3. Asigne el eje $\hat{Z}_i$ para que apunte sobre el i-ésimo eje de articulación.
4. Asigne el eje $\hat{X}_i$ para que apunte sobre la perpendicular común o, si los ejes se intersecan, asigne $\hat{X}_i$ para que sea normal al plano que contiene los dos ejes.
5. Asigne el eje $\hat{Y}_i$ para completar un sistema de coordenadas de mano derecha.
6. Asigne $\{0\}$ para que concuerde con $\{1\}$ cuando la primera variable de articulación sea cero. Para $\{N\}$ seleccione la ubicación del origen y la dirección de $\hat{X}_N$ libremente, pero generalmente de manera que haga que la mayor parte de los parámetros de los vínculos sean cero.

EJEMPLO 3.3

La figura 3.6(a) muestra un brazo planar de tres vínculos. Como las tres articulaciones son angulares, a este manipulador algunas veces se le conoce como **mecanismo *RRR*** (o **3*R***). La figura 3.6(b) es una representación esquemática del mismo manipulador.

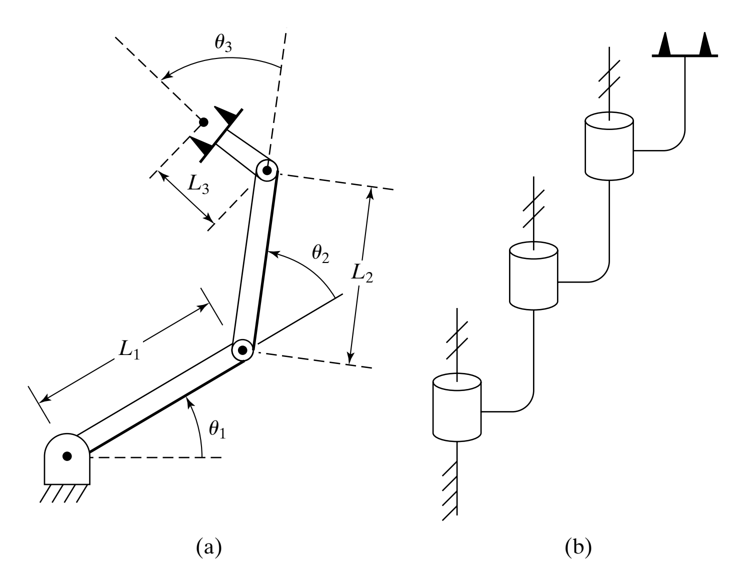


FIGURA 3.6: Un brazo planar de tres vínculos. A la derecha mostramos el mismo manipulador por medio de una notación esquemática simple. Las marcas de referencia en los ejes indican que son mutuamente paralelos.

Observe las marcas de referencia dobles en cada uno de los tres ejes para indicar que son paralelos. Asigne tramas de vínculo a este mecanismo y proporcione los parámetros de Denavit-Hartenberg.

Empezamos definiendo la trama de referencia, a la que llamaremos trama {0}. Está fija a la base y se alinea con la trama {1} cuando la primera variable de articulación (θ_1) es cero. Por lo tanto, posicionamos la trama {0} como se muestra en la figura 3.7, con el eje $\hat{Z}_0$ alineado con el eje de la articulación 1. Para este brazo, todos los ejes de

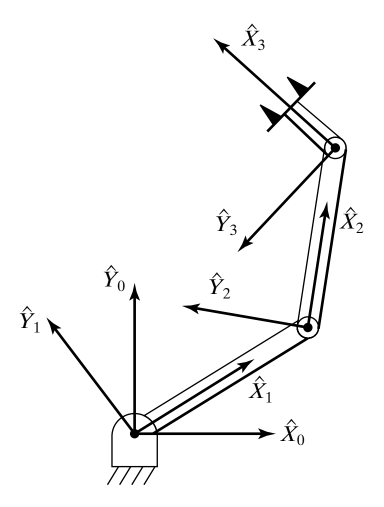


FIGURA 3.7: Asignaciones de tramas a los vínculos.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	L_1	0	θ_2
3	0	L_2	0	θ_3

FIGURA 3.8: Parámetros de vínculo del manipulador planar de tres vínculos.

articulación están orientados en forma perpendicular al plano del brazo. Como el brazo se encuentra en un plano con todos los ejes $\hat{Z}$ paralelos, no hay desplazamientos de vínculo; todas las d_i son cero. Todas las articulaciones son giratorias, por lo que cuando están a cero grados todos los ejes $\hat{X}$ deben alinearse.

Teniendo en cuenta estos comentarios, es fácil encontrar las asignaciones de tramas que se muestran en la figura 3.7. Los parámetros de vínculo correspondientes se muestran en la figura 3.8.

Observe que, como los ejes de articulación son todos paralelos y todos los ejes $\hat{Z}$ se consideran como apuntando hacia fuera del papel, todas las α_i son cero. Éste es obviamente un mecanismo muy simple. Observe también que nuestro análisis cinemático siempre termina en una trama cuyo origen se encuentra en el último eje de articulación; por lo tanto, l_3 no aparece en los parámetros de vínculo. Dichos desplazamientos finales hacia el efector final se tratarán por separado más adelante.

EJEMPLO 3.4

La figura 3.9(a) muestra un robot que tiene tres grados de libertad y una articulación prismática. A este manipulador se le puede llamar "mecanismo *RPR*", en una notación que especifique el tipo y orden de las articulaciones. Es un robot "cilíndrico" cuyas primeras dos articulaciones son análogas a las coordenadas polares cuando se les ve desde arriba. La última articulación (3) proporciona el "giro" para la mano. La figura 3.9(b) muestra el mismo manipulador en forma esquemática. Observe el símbolo utilizado

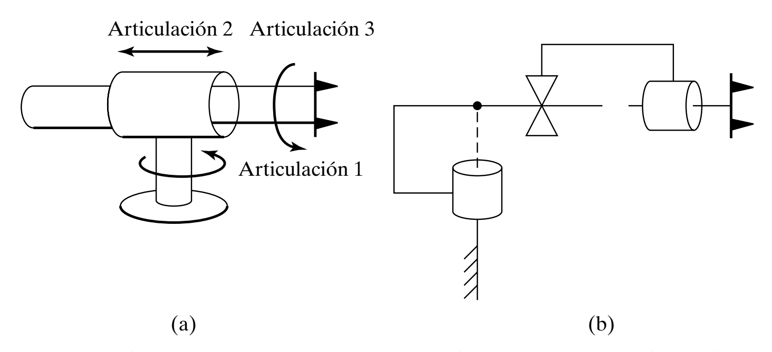


FIGURA 3.9: Un manipulador con tres grados de libertad y una articulación prismática.

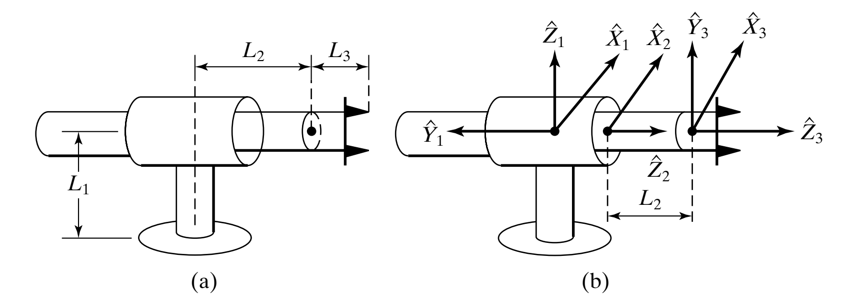


FIGURA 3.10: Asignaciones tramas a los vínculos.

para representar articulaciones prismáticas y también que se utiliza un "punto" para indicar el lugar en el que se intersecan dos ejes adyacentes. Además, se ha indicado el hecho de que los ejes 1 y 2 son ortogonales.

La figura 3.10(a) muestra al manipulador con la articulación prismática en su mínima extensión; la asignación de tramas a los vínculos se muestra en la figura 3.10(b).

Observe que las tramas {0} y {1} se muestran como exactamente coincidentes en esta figura, ya que el robot se dibuja para la posición $\theta_1 = 0$. Observe que la trama {0}, aunque no se encuentra en la parte inferior de la base bridada del robot, está unida rígidamente al vínculo 0, la parte inmóvil del robot. Así como nuestras tramas de vínculos no se utilizan para describir el desplazamiento hasta llegar a la mano, no necesitan unirse completamente hasta la parte más baja de la base del robot. Es suficiente que se una la trama {0} en cualquier parte al vínculo fijo 0, y que la trama {N} (la trama final) se una en cualquier parte al último vínculo del manipulador. Otros desplazamientos podrán manejarse posteriormente de una manera general.

Note que las articulaciones giran sobre el eje $\hat{Z}$ de la trama asociada, pero las articulaciones prismáticas se deslizan a lo largo de $\hat{Z}$. En el caso en que la articulación i sea prismática, θ_i es una constante fija y d_i es la variable. Si d_i es cero en la mínima extensión del vínculo, entonces la trama {2} deberá unirse en donde se muestra, para que d_2 proporcione el verdadero desplazamiento. Los parámetros de vínculo se muestran en la figura 3.11.

Observe que θ_2 es cero para este robot y que d_2 es una variable. Los ejes 1 y 2 se intersecan, por lo que a_1 es cero. El ángulo α_1 debe ser de 90 grados para poder girar $\hat{Z}_1$ de manera que se alinee con $\hat{Z}_2$ (sobre $\hat{X}_1$).

EJEMPLO 3.5

La figura 3.12(a) muestra un manipulador $3R$ de tres vínculos, en el que los ejes de articulación 1 y 2 se intersecan y los ejes 2 y 3 son paralelos. La figura 3.12(b) muestra el esquema cinemático del manipulador. Observe que el esquema incluye anotaciones que indican que los primeros dos ejes son ortogonales y que los últimos dos son paralelos.

Demuestre la no unicidad de las asignaciones de tramas y de los parámetros de Denavit-Hartenberg, mostrando varias posibles asignaciones correctas de las tramas {1} y {2}.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	90°	0	d_2	0
3	0	0	L_2	θ_3

FIGURA 3.11: Parámetros de vínculo para el manipulador *RPR* del ejemplo 3.4.

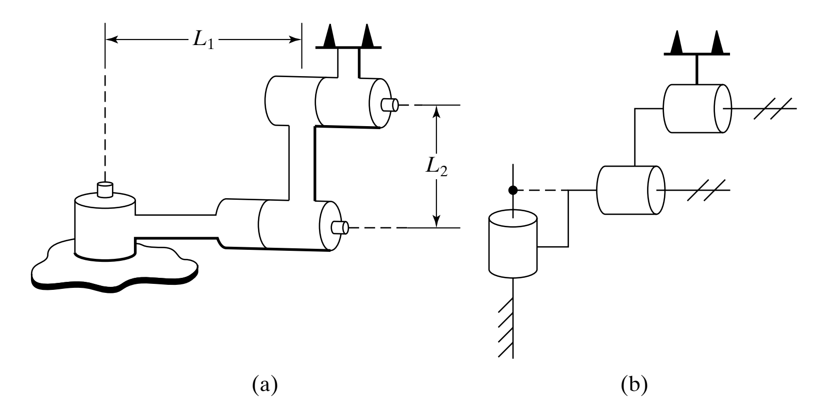


FIGURA 3.12: Manipulador no planar de tres vínculos.

La figura 3.13 muestra dos posibles asignaciones de tramas y los parámetros correspondientes para las dos posibles opciones de dirección de $\hat{Z}_2$.

En general, cuando $\hat{Z}_i$ y $\hat{Z}_{i+1}$ se intersecan hay dos opciones para X_i. En este ejemplo, los ejes de articulación 1 y 2 se intersecan, por lo que hay dos opciones para la dirección de $\hat{X}_1$. La figura 3.14 muestra otras dos posibles asignaciones de tramas que corresponden a la segunda opción de $\hat{X}_1$.

De hecho hay cuatro posibilidades más que corresponden a las cuatro opciones anteriores, pero con $\hat{Z}_1$ apuntando hacia abajo.

3.5 CINEMÁTICA DE MANIPULADORES

En esta sección derivaremos la forma general de la transformación que relaciona las tramas unidas a vínculos adyacentes. Luego concatenaremos estas transformaciones individuales para despejar la posición y orientación del vínculo n relativo al vínculo 0.

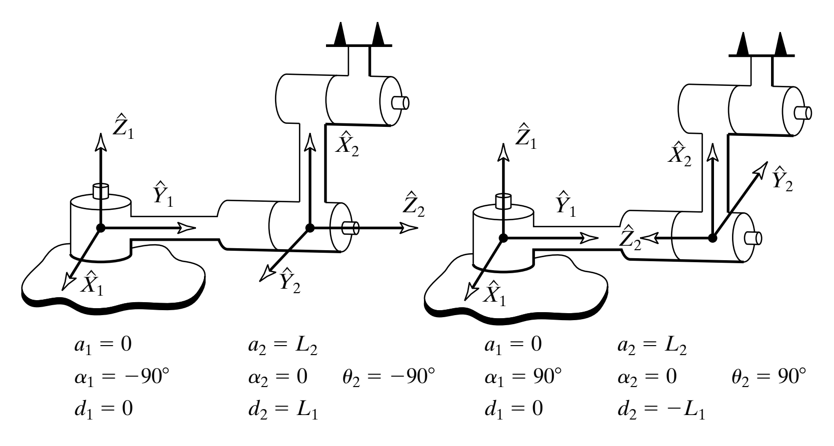


FIGURA 3.13: Dos posibles asignaciones de tramas.

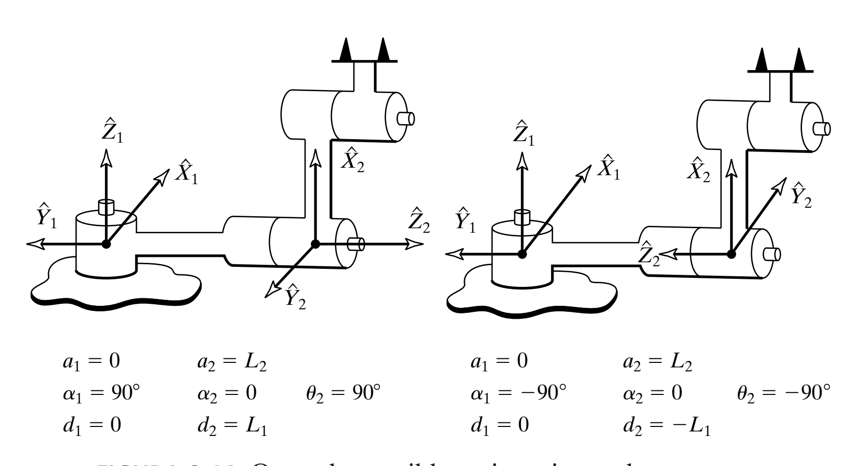


FIGURA 3.14: Otras dos posibles asignaciones de tramas.

Derivación de transformaciones de vínculos

Deseamos construir la transformada que define la trama $\{i\}$ relativa a la trama $\{i-1\}$. En general, esta transformación será una función de los cuatro parámetros de vínculo. Para cualquier robot *dado*, esta transformación será una función de una variable solamente, y los otros tres parámetros serán fijos por diseño mecánico. Al definir una trama para cada vínculo hemos descompuesto el problema de cinemática en n subproblemas. Para poder resolver cada uno de estos subproblemas, específicamente ${}^{i-1}_{\ i}T$, los desglosaremos aún más en cuatro "subsubproblemas". *Cada una de estas transformaciones será una función de un parámetro de vínculo solamente, y será lo suficientemente simple como para que podamos anotar su forma por inspección*. Empezaremos definiendo tres tramas intermedias para cada vínculo: $\{P\}$, $\{Q\}$ y $\{R\}$.

La figura 3.15 muestra el mismo par de articulaciones de antes, con las tramas $\{P\}$, $\{Q\}$ y $\{R\}$ definidas. Observe que sólo se muestran los ejes $\hat{X}$ y $\hat{Z}$ de cada trama, para que el dibujo sea más claro. La trama $\{R\}$ difiere de la trama $\{i-1\}$ sólo por una rotación de α_{i-1}.

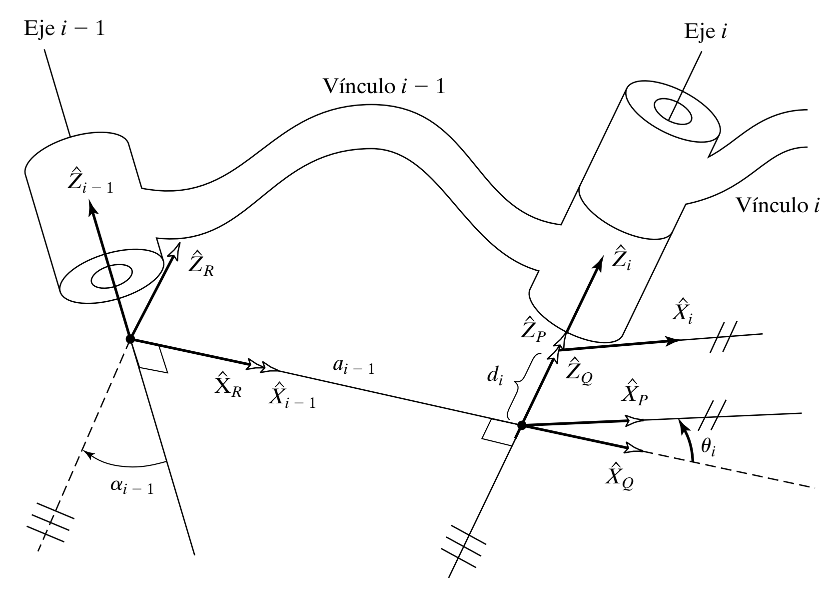


FIGURA 3.15: Ubicación de tramas intermedias $\{P\}$, $\{Q\}$ y $\{R\}$.

La trama $\{Q\}$ difiere de $\{R\}$ por una traslación a_{i-1}. La trama $\{P\}$ difiere de $\{Q\}$ por una rotación θ_i, y la trama $\{i\}$ difiere de $\{P\}$ por una traslación d_i. Si deseamos escribir la transformación que modifica los vectores definidos en $\{i\}$ a su descripción en $\{i-1\}$, podemos hacerlo de la siguiente forma

$$ {}^{i-1}P = {}^{i-1}_{R}T \; {}^{R}_{Q}T \; {}^{Q}_{P}T \; {}^{P}_{i}T \; {}^{i}P, \tag{3.1}$$

o

$$ {}^{i-1}P = {}^{i-1}_{i}T \; {}^{i}P, \tag{3.2}$$

en donde

$$ {}^{i-1}_{i}T = {}^{i-1}_{R}T \; {}^{R}_{Q}T \; {}^{Q}_{P}T \; {}^{P}_{i}T. \tag{3.3}$$

Considerando cada una de estas transformaciones, podemos ver que la ecuación (3.3) puede escribirse así:

$$ {}^{i-1}_{i}T = R_X(\alpha_{i-1}) D_X(a_{i-1}) R_Z(\theta_i) D_Z(d_i), \tag{3.4}$$

o

$$ {}^{i-1}_{i}T = \text{Tornillo}_X(a_{i-1}, \alpha_{i-1}) \; \text{Tornillo}_Z(d_i, \theta_i), \tag{3.5}$$

en donde la notación $\text{Tornillo}_Q(r, \phi)$ representa la combinación de una traslación sobre un eje $\hat{Q}$ por una distancia r y una rotación sobre el mismo eje por un ángulo ϕ. Realizando la multiplicación en la ecuación (3.4) obtenemos la forma general de ${}^{i-1}_{i}T$:

$$ {}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.6}$$

EJEMPLO 3.6

Utilizando los parámetros de vínculo mostrados en la figura 3.11 para el robot de la figura 3.9, calcule las transformaciones individuales para cada vínculo.

Sustituyendo los parámetros en la ecuación (3.6), obtenemos

$$
\begin{aligned}
{}^{0}_{1}T &= \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
{}^{1}_{2}T &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -d_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
{}^{2}_{3}T &= \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & 0 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned}
\qquad (3.7)
$$

Una vez que hayamos derivado estas transformaciones de vínculos, será conveniente comprobarlos con sentido común. Por ejemplo, los elementos de la cuarta columna de cada transformación deben dar las coordenadas del origen de la siguiente trama de mayor orden.

Concatenación de las transformaciones de vínculos

Una vez que se han definido las tramas de vínculo y se han encontrado los correspondientes parámetros de vínculo, es muy simple desarrollar las ecuaciones cinemáticas. Las matrices de transformación de vínculos individuales pueden calcularse a partir de los valores de los parámetros de vínculo. Después pueden multiplicarse las transformaciones de vínculo entre sí para encontrar la transformación que relaciona la trama $\{N\}$ con la trama $\{0\}$:

$$ {}^{0}_{N}T = {}^{0}_{1}T\, {}^{1}_{2}T\, {}^{2}_{3}T \ldots {}^{N-1}_{N}T. \qquad (3.8) $$

Esta transformación ${}^{0}_{N}T$ será una función de todas las n variables de articulación. Si se consultan los sensores de posición de articulación del robot, puede calcularse la posición y orientación cartesiana del último vínculo mediante ${}^{0}_{N}T$.

3.6 ESPACIO DE ACTUADOR, ESPACIO DE ARTICULACIÓN Y ESPACIO CARTESIANO

La posición de todos los vínculos de un manipulador con n grados de libertad puede especificarse con un conjunto de n variables de articulación. Este conjunto de variables a menudo se conoce como el **vector de articulación** de $n \times 1$. El espacio de dichos vectores de articulación se conoce como **espacio de articulación**. Hasta ahora nos hemos preocupado por calcular la descripción de **espacio cartesiano** a partir de la descripción del espacio de articulación. Utilizamos el término *espacio cartesiano* cuando la posición se mide a lo largo de ejes ortogonales y la orientación se mide de acuerdo con cualquiera de las convenciones descritas en el capítulo 2. Algunas veces se utilizan los términos **espacio orientado a tarea** y **espacio operacional** para lo que llamaremos espacio cartesiano.

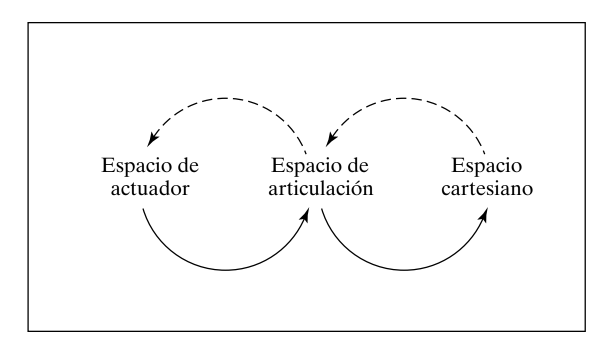


FIGURA 3.16: Asignaciones entre descripciones cinemáticas.

Hasta ahora hemos asumido implícitamente que cada articulación cinemática es accionada directamente por algún tipo de actuador. No obstante, en el caso de muchos robots industriales, esto no es así. Por ejemplo, algunas veces dos actuadores trabajan juntos en un par diferencial para mover una sola articulación; otras veces se utiliza un actuador lineal para girar una articulación angular mediante el uso de una vinculación de cuatro barras. En estos casos es útil considerar la noción de *posiciones de actuador*. Los sensores que miden la posición del manipulador a menudo se encuentran ubicados en los actuadores, por lo que deben realizarse ciertos cálculos para determinar el vector de articulación como una función del conjunto de valores de un actuador, o **vector de actuador**.

Como se indica en la figura 3.16, hay tres representaciones de la posición y la orientación de un manipulador: descripciones en **espacio de actuador**, en **espacio de articulación** y en **espacio cartesiano**. En este capítulo nos interesan las asignaciones entre representaciones, según lo indican las flechas sólidas en la figura 3.16. En el capítulo 4 consideraremos las asignaciones inversas, indicadas por las flechas punteadas.

Hay muchas formas en que los actuadores se pueden conectar para mover una articulación; podrían catalogarse, pero no haremos eso aquí. Para cada robot que diseñemos o busquemos analizar, debemos averiguar la correspondencia entre las posiciones de los actuadores y las posiciones de las articulaciones. En la siguiente sección resolveremos un problema de ejemplo para un robot industrial.

3.7 EJEMPLOS: CINEMÁTICA DE DOS ROBOTS INDUSTRIALES

Los robots industriales actuales están disponibles en muchas configuraciones cinemáticas [2], [3]. En esta sección trabajaremos con el comportamiento cinemático de dos robots industriales típicos. Primero consideraremos el Unimation PUMA 560, un manipulador de articulación giratoria con seis grados de libertad. Resolveremos las ecuaciones cinemáticas como funciones de los ángulos de articulación. Para este ejemplo omitiremos el problema adicional de la relación entre el espacio de actuador y el espacio de articulación. En segundo lugar consideraremos el Yasukawa Motoman L-3, un robot con cinco grados de libertad y articulaciones giratorias. Realizaremos este ejemplo con todo detalle, incluyendo las transformaciones de actuador a articulación. Si es la primera vez que lee este libro, puede omitir este ejemplo.

El PUMA 560

El Unimation PUMA 560 (figura 3.17) es un robot con seis grados de libertad y todas las articulaciones giratorias (es decir, es un mecanismo 6R). En la figura 3.18 se muestra



FIGURA 3.17: El Unimation PUMA 560. Cortesía de Unimation Incorporated, Shelter Rock Lane, Danbury, Conn.

con asignaciones de tramas en la posición para la que todos los ángulos de articulación son iguales a cero.[5] La figura 3.19 muestra los detalles del antebrazo del robot.

Observe que la trama {0} (que no se muestra) es coincidente con la trama {1} cuando θ_1 es cero. Observe también que, para este robot, al igual que para muchos robots industriales, los ejes de las articulaciones 4, 5 y 6 se intersecan todos en un punto común, y este punto de intersección coincide con el origen de las tramas {4}, {5} y {6}. Lo que es más, los ejes de articulación 4, 5 y 6 son mutuamente ortogonales. Este mecanismo de muñeca se muestra esquemáticamente en la figura 3.20.

Los parámetros de vínculo que corresponden a esta disposición de tramas de vínculos se muestran en la figura 3.21. En el caso del PUMA 560, una combinación de engranajes en la muñeca del manipulador acopla los movimientos de las articulaciones 4, 5 y 6. Esto significa que para estas tres articulaciones debemos hacer una distinción entre espacio de articulación y espacio de actuador, y resolver la cinemática completa en dos pasos. No obstante, en este ejemplo consideraremos solamente la cinemática del espacio de articulación al espacio cartesiano.

[5] Unimation ha utilizado una asignación ligeramente distinta de la ubicación cero de las articulaciones, de tal forma que $\theta_3^* = \theta_3 - 180°$, en donde θ_3^* es la posición de la articulación 3 en la convención de Unimation.

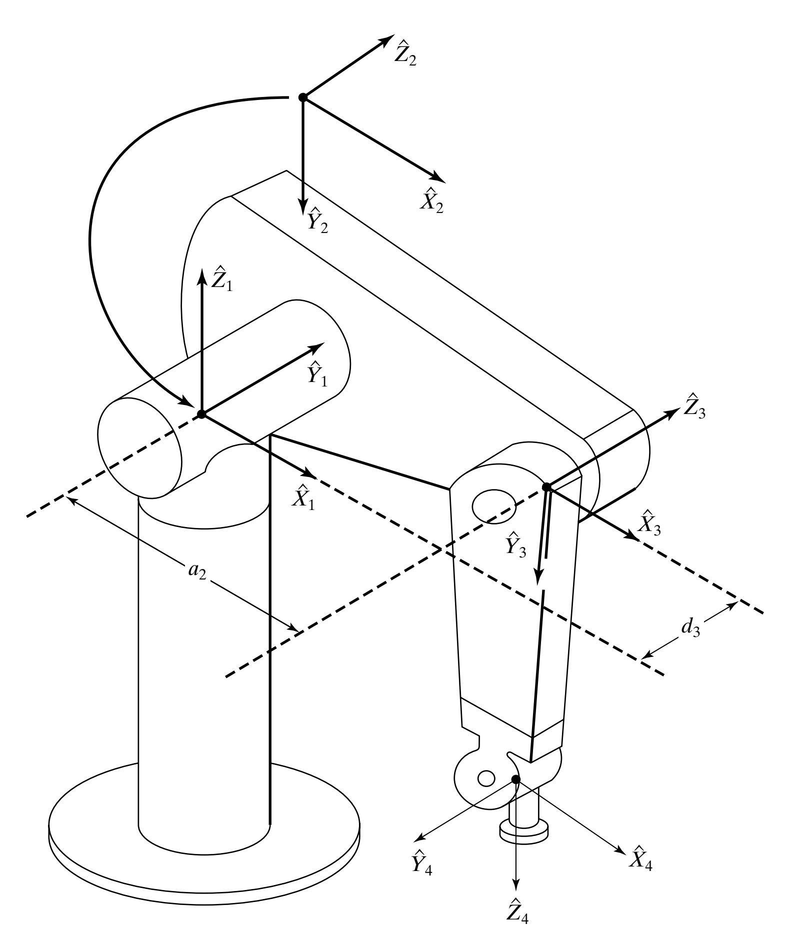


FIGURA 3.18: Algunos parámetros de cinemática y asignaciones de tramas para el manipulador PUMA 560.

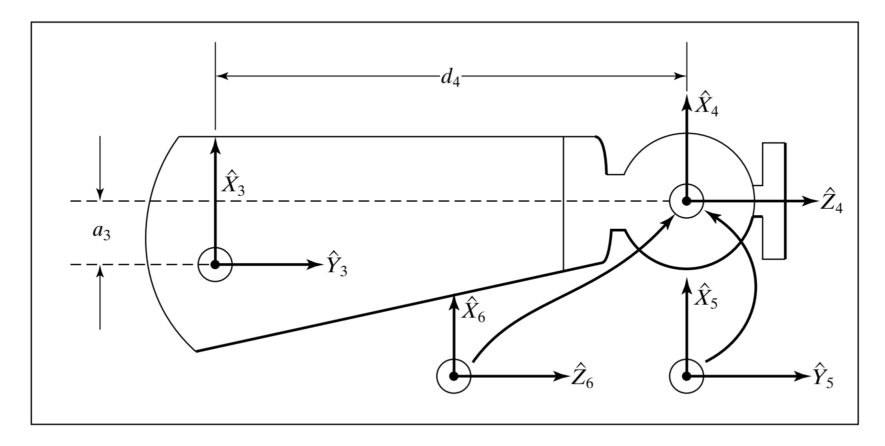


FIGURA 3.19: Parámetros cinemáticos y asignaciones de tramas para el antebrazo del manipulador PUMA 560.

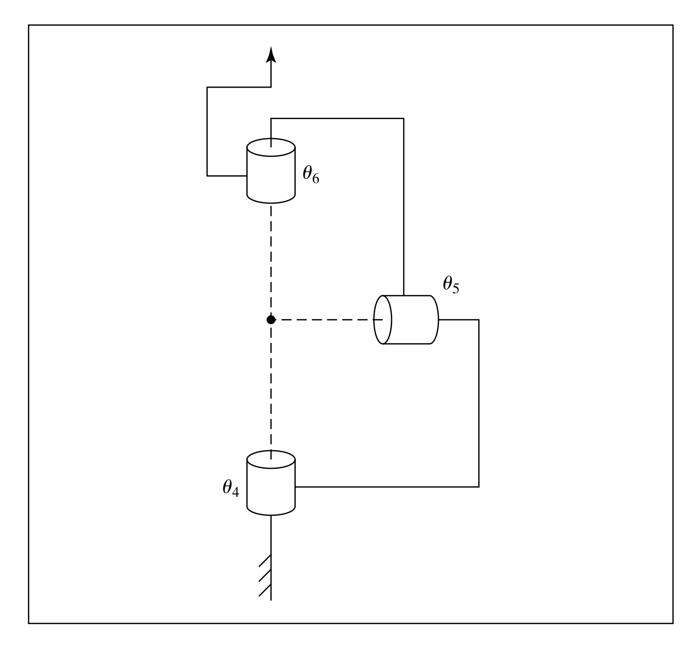


FIGURA 3.20: Diagrama esquemático de una muñeca 3R en la que los tres ejes se intersecan en un punto y son mutuamente ortogonales. Este diseño se utiliza en el manipulador PUMA 560 y en muchos otros robots industriales.

i	$\alpha_i - 1$	$a_i - 1$	d_i	θi
1	0	0	0	θ_1
2	$-90°$	0	0	θ_2
3	0	a_2	d_3	θ_3
4	$-90°$	a_3	d_4	θ_4
5	$90°$	0	0	θ_5
6	$-90°$	0	0	θ_6

FIGURA 3.21: Parámetros de vínculo del PUMA 560.

Utilizando la ecuación (3.6) calculamos cada una de las transformaciones de vínculo:

$$
{}^0_1T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^1_2T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_2 & -c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^2_3T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & a_2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^3_4T = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ -s\theta_4 & -c\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^4_5T = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^5_6T = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_6 & -c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.9}
$$

Ahora formamos ${}^0_6T$ mediante la multiplicación matricial de las matrices de vínculo individuales. Mientras formamos este producto, derivaremos algunos subresultados que serán útiles para resolver el problema de cinemática inversa que se presenta en el capítulo 4. Empezaremos multiplicando ${}^4_5T$ y ${}^5_6T$; esto es:

$$
{}^4_6T = {}^4_5T\,{}^5_6T = \begin{bmatrix} c_5c_6 & -c_5s_6 & -s_5 & 0 \\ s_6 & c_6 & 0 & 0 \\ s_5c_6 & -s_5s_6 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.10}
$$

en donde c_5 es abreviación de cos θ_5, s_5 de sen θ_5 y así sucesivamente.[6] Así, tenemos que

$$
{}^3_6T = {}^3_4T\,{}^4_6T = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & -c_4s_5 & a_3 \\ s_5c_6 & -s_5s_6 & c_5 & d_4 \\ -s_4c_5c_6 - c_4s_6 & s_4c_5s_6 - c_4c_6 & s_4s_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.11}
$$

[6] Dependiendo de la cantidad de espacio disponible para mostrar las expresiones, utilizaremos cualquiera de las siguientes tres formas: cos θ_5, $c\theta_5$ o c_5.

Como las articulaciones 2 y 3 son siempre paralelas, si multiplicamos primero ${}^{1}_{2}T$ y ${}^{2}_{3}T$ y luego aplicamos fórmulas para sumar ángulos, obtendremos una expresión final más simple. Esto puede hacerse siempre que dos articulaciones giratorias tengan ejes paralelos; así, tenemos que

$$ {}^{1}_{3}T = {}^{1}_{2}T \, {}^{2}_{3}T = \begin{bmatrix} c_{23} & -s_{23} & 0 & a_2c_2 \\ 0 & 0 & 1 & d_3 \\ -s_{23} & -c_{23} & 0 & -a_2s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.12} $$

en donde hemos usado las fórmulas de suma de ángulos (del apéndice A):

$$ c_{23} = c_2c_3 - s_2s_3, $$
$$ s_{23} = c_2s_3 + s_2c_3. $$

Después tenemos que

$$ {}^{1}_{6}T = {}^{1}_{3}T \, {}^{3}_{6}T = \begin{bmatrix} {}^{1}r_{11} & {}^{1}r_{12} & {}^{1}r_{13} & {}^{1}p_x \\ {}^{1}r_{21} & {}^{1}r_{22} & {}^{1}r_{23} & {}^{1}p_y \\ {}^{1}r_{31} & {}^{1}r_{32} & {}^{1}r_{33} & {}^{1}p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

en donde

$$ \begin{aligned} {}^{1}r_{11} &= c_{23}[c_4c_5c_6 - s_4s_6] - s_{23}s_5s_6, \\ {}^{1}r_{21} &= -s_4c_5c_6 - c_4s_6, \\ {}^{1}r_{31} &= -s_{23}[c_4c_5c_6 - s_4s_6] - c_{23}s_5c_6, \\ {}^{1}r_{12} &= -c_{23}[c_4c_5s_6 + s_4c_6] + s_{23}s_5s_6, \\ {}^{1}r_{22} &= s_4c_5s_6 - c_4c_6, \\ {}^{1}r_{32} &= s_{23}[c_4c_5s_6 + s_4c_6] + c_{23}s_5s_6, \\ {}^{1}r_{13} &= -c_{23}c_4s_5 - s_{23}c_5, \\ {}^{1}r_{23} &= s_4s_5, \\ {}^{1}r_{33} &= s_{23}c_4s_5 - c_{23}c_5, \\ {}^{1}p_x &= a_2c_2 + a_3c_{23} - d_4s_{23}, \\ {}^{1}p_y &= d_3, \\ {}^{1}p_x &= -a_3s_{23} - a_2s_2 - d_4c_{23}. \end{aligned} \tag{3.13} $$

Finalmente obtenemos el producto de las seis transformadas de vínculos:

$$ {}^{0}_{6}T = {}^{0}_{1}T \, {}^{1}_{6}T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

Aquí,

$$r_{11} = c_1[c_{23}(c_4c_5c_6 - s_4s_5) - s_{23}s_5c_5] + s_1(s_4c_5c_6 + c_4s_6),$$
$$r_{21} = s_1[c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6 - c_1(s_4c_5c_6 + c_4s_6),$$
$$r_{31} = -s_{23}(c_4c_5c_6 - s_4s_6) - c_{23}s_5c_6,$$

$$r_{12} = c_1[c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6] + s_1(c_4c_6 - s_4c_5s_6),$$
$$r_{22} = s_1[c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6] - c_1(c_4c_6 - s_4c_5s_6),$$
$$r_{32} = -s_{23}(-c_4c_5s_6 - s_4c_6) + c_{23}s_5s_6,$$

$$r_{13} = -c_1(c_{23}c_4s_5 + s_{23}c_5) - s_1s_4s_5,$$
$$r_{23} = -s_1(c_{23}c_4s_5 + s_{23}c_5) + c_1s_4s_5,$$
$$r_{33} = s_{23}c_4s_5 - c_{23}c_5,$$

$$p_x = c_1[a_2c_2 + a_3c_{23} - d_4s_{23}] - d_3s_1,$$
$$p_y = s_1[a_2c_2 + a_3c_{23} - d_4s_{23}] + d_3c_1,$$
$$p_z = -a_3s_{23} - a_2s_2 - d_4c_{23}. \qquad (3.14)$$

Las ecuaciones (3.14) constituyen la cinemática del PUMA 560. Especifican cómo calcular la posición y la orientación de la trama {6} relativa a la trama {0} del robot. Éstas son las ecuaciones básicas para todos los análisis cinemáticos de este manipulador.

El Yasukawa Motoman L-3

Este robot es un manipulador industrial popular con cinco grados de libertad (figura 3.22). A diferencia de los ejemplos que hemos visto hasta ahora, el Motoman no es una simple cadena cinemática abierta sino que utiliza dos actuadores lineales acoplados a los vínculos 2 y 3, con articulaciones de 4 barras. Además, las articulaciones 4 y 5 se operan a través de una transmisión de cadena por dos actuadores, en un arreglo diferencial.

En este ejemplo resolveremos su comportamiento cinemático en dos etapas. Primero despejaremos los ángulos de articulación en términos de las posiciones de los actuadores; después despejaremos la posición y orientación cartesiana del último vínculo en términos de los ángulos de articulación. En esta segunda etapa podemos tratar el sistema como si fuera un dispositivo 5*R* de cadena cinemática abierta simple.

La figura 3.23 muestra el mecanismo articulado que conecta el actuador número 2 con los vínculos 2 y 3 del robot. El actuador es lineal y controla directamente la longitud del segmento etiquetado como DC. El triángulo ABC es fijo, al igual que la longitud BD. La articulación 2 pivotea sobre el punto B y el actuador pivotea ligeramente sobre el punto C al moverse la articulación. Daremos los siguientes nombres a las constantes (longitudes y ángulos) asociadas con el actuador 2:

$$\gamma_2 = AB, \phi_2 = AC, \alpha_2 = BC,$$

$$\beta_2 = BD, \Omega_2 = \angle JBD, l_2 = BJ,$$

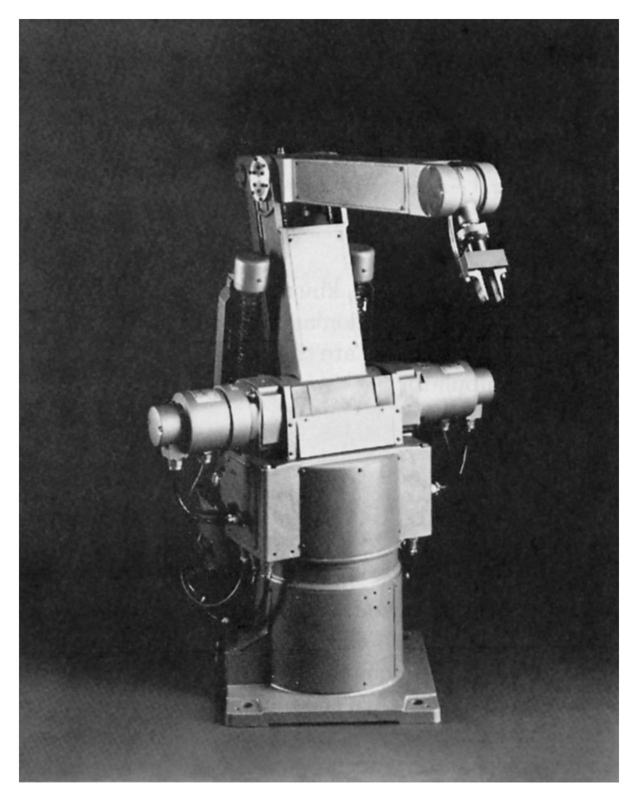


FIGURA 3.22: El Yasukawa Motoman L-3. Cortesía de Yasukawa.

daremos los siguientes nombres a las variables:

$$\theta_2 = -\angle JBQ, \psi_2 = \angle CBD, g_2 = DC.$$

La figura 3.24 muestra el mecanismo de unión que conecta al actuador número 3 con los vínculos 2 y 3 del robot. El actuador es lineal y controla directamente la longitud del segmento etiquetado como *HG*. El triángulo *EFG* es fijo, al igual que la longitud *FH*. La articulación 3 pivotea sobre el punto *J* y el actuador pivotea ligeramente sobre el punto *G*, a medida que se mueve la articulación. Daremos los siguientes nombres a las constantes (longitudes y ángulos) asociadas con el actuador 3:

$$\gamma_3 = EF, \phi_3 = EG, \alpha_3 = GF,$$
$$\beta_3 = HF, l_3 = JK.$$

Daremos los siguientes nombres a las variables:

$$\theta_3 = \angle PJK, \ \psi_3 = \angle GFH, \ g_3 = GH.$$

Este arreglo de actuadores y articulaciones tiene el siguiente efecto funcional. El actuador 2 se utiliza para posicionar la articulación 2; mientras hace esto, el vínculo 3 permanece en la misma orientación relativa a la base del robot. El actuador 3 se utiliza

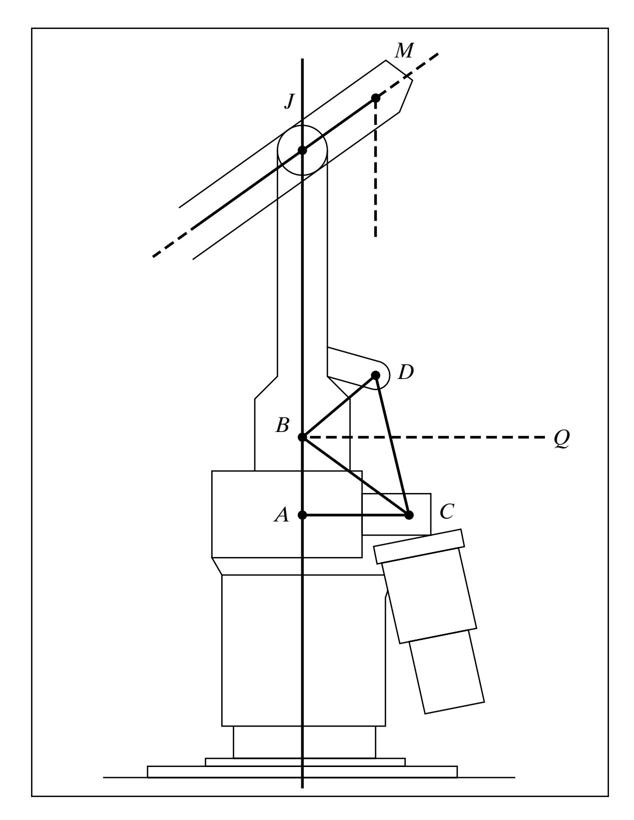


FIGURA 3.23: Detalles cinemáticos de la articulación del actuador 2 de Yasukawa.

para ajustar la orientación del vínculo 3 respecto a la base del robot (en vez de ser respecto al vínculo anterior, como en un robot de cadena cinemática serial). Un propósito de dicho arreglo de articulaciones es incrementar la rigidez estructural de las articulaciones principales del robot. Esto a menudo recompensa en términos de una mayor habilidad para posicionar el robot con precisión.

Los actuadotes para las articulaciones 4 y 5 se unen al vínculo 1 del robot con sus ejes alineados con los de la articulación 2 (puntos *B* y *F* en las figuras 3.23 y 3.24). Operan las articulaciones de la muñeca mediante dos conjuntos de cadenas: uno localizado de manera interior al vínculo 2, el segundo localizado de manera interior al vínculo 3. El efecto de este sistema de transmisión, junto con su interacción con el accionamiento de los vínculos 2 y 3, se describe funcionalmente de la siguiente manera: el actuador 4 se utiliza para posicionar la articulación 4 relativa a la base del robot, en vez de ser relativa al vínculo 3 anterior. Esto significa que si el actuador 4 se mantiene constante, el vínculo 4 se mantendrá a una orientación constante con respecto a la base del robot, sin importar las posiciones de las articulaciones 2 y 3. Finalmente, el actuador 5 se comporta como si estuviera directamente conectado a la articulación 5.

Ahora definiremos las ecuaciones que asignan un conjunto de valores de actuador (A_i) al conjunto equivalente de valores de articulación (θ_i). En este caso, estas ecuaciones se derivaron mediante geometría plana elemental; la mayoría con la sola aplicación de la “ley de los cosenos.”[7]

[7] Si los ángulos de un triángulo son *a*, *b* y *c*, en donde el ángulo θ se encuentra opuesto al lado *A*, y así sucesivamente, entonces $A^2 = B^2 + C^2 - 2BC \cos a$.

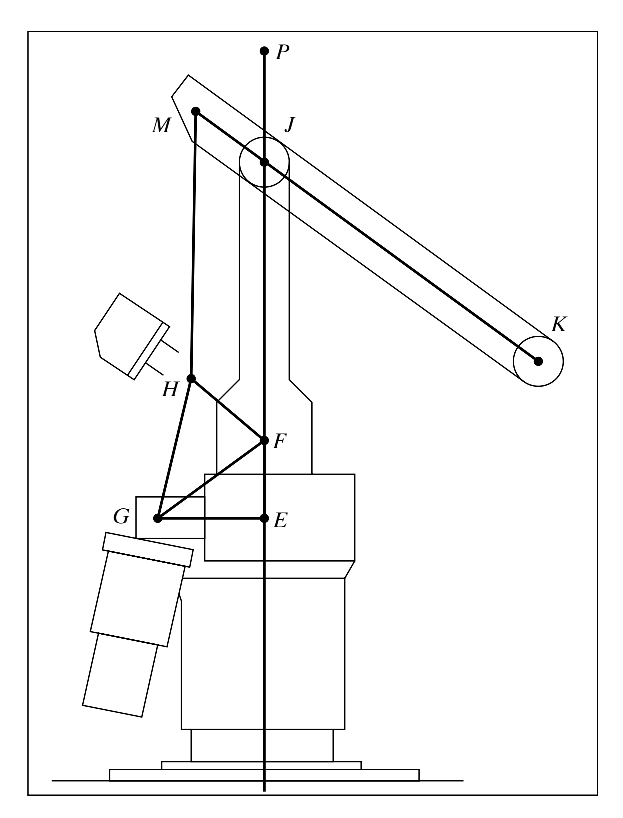


FIGURA 3.24: Detalles cinemáticos de la articulación del actuador 3 de Yasukawa.

En estas ecuaciones aparecen las constantes de escala (k_i) y desplazamiento (λ_i) para cada actuador. Por ejemplo, el actuador 1 está conectado directamente al eje de articulación 1, por lo que la conversión es simple; se trata únicamente de un factor de escala más un desplazamiento. Por lo tanto,

$$
\begin{aligned}
\theta_1 &= k_1 A_1 + \lambda_1, \\
\theta_2 &= \cos^{-1}\left(\frac{(k_2 A_2 + \lambda_2)^2 - \alpha_2^2 - \beta_2^2}{-2\alpha_2\beta_2}\right) + \tan^{-1}\left(\frac{\phi_2}{\gamma_2}\right) + \Omega_2 - 270^\circ, \\
\theta_3 &= \cos^{-1}\left(\frac{(k_3 A_3 + \lambda_3)^2 - \alpha_3^2 - \beta_3^2}{-2\alpha_3\beta_3}\right) - \theta_2 + \tan^{-1}\left(\frac{\phi_3}{\gamma_3}\right) - 90^\circ, \\
\theta_4 &= -k_4 A_4 - \theta_2 - \theta_3 + \lambda_4 + 180^\circ, \qquad (3.15) \\
\theta_5 &= -k_5 A_5 + \lambda_5.
\end{aligned}
$$

La figura 3.25 muestra la asignación de tramas a los vínculos. En esta figura el manipulador se muestra en una posición correspondiente al vector de articulación $\Theta = (0, -90^\circ, 90^\circ, 90^\circ, 0)$. La figura 3.26 muestra los parámetros de vínculo para este manipulador. Las matrices de transformación de vínculos que se obtienen son

$$
{}^{0}_{1}T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{1}_{2}T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_2 & -c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{2}_{3}T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & l_2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.16}
$$

$$
{}^{3}_{4}T = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & l_3 \\ s\theta_4 & c\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{4}_{5}T = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

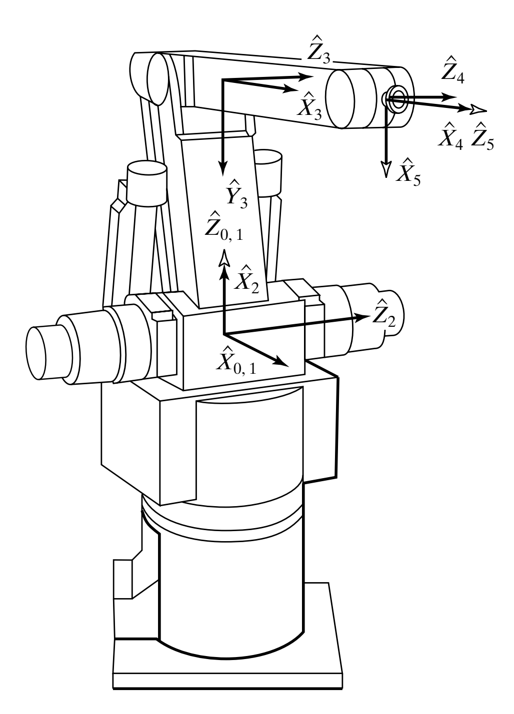


FIGURA 3.25: Asignación de las tramas de vínculo para el Yasukawa L-3.

i	$\alpha_i - 1$	$a_i - 1$	d_i	θ_i
1	0	0	0	θ_1
2	$-90°$	0	0	θ_2
3	0	L_2	0	θ_3
4	0	L_3	0	θ_4
5	$90°$	0	0	θ_5

FIGURA 3.26: Parámetros de vínculo del manipulador Yasukawa L-3.

Formando el producto para obtener ${}^0_5T$, obtenemos

$$
{}^0_5T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

en donde

$$
\begin{aligned}
r_{11} &= c_1 c_{234} c_5 - s_1 s_5, \\
r_{21} &= s_1 c_{234} c_5 + c_1 s_5, \\
r_{31} &= -s_{234} c_5, \\
r_{12} &= -c_1 c_{234} s_5 - s_1 c_5, \\
r_{22} &= -s_1 c_{234} s_5 + c_1 c_5, \\
r_{32} &= s_{234} s_5, \\
r_{13} &= c_1 s_{234}, \\
r_{23} &= s_1 s_{234}, \\
r_{33} &= c_{234},
\end{aligned}
$$

$$p_x = c_1(l_2c_2 + l_3c_{23}),$$
$$p_y = s_1(l_2c_2 + l_3c_{23}),$$
$$p_z = -l_2s_2 - l_3s_{23}. \tag{3.17}$$

Desarrollamos las ecuaciones cinemáticas para el Yasukawa Motoman en dos pasos: en el primero calculamos un vector de articulación a partir de un vector de actuador; en el segundo calculamos una posición y orientación de la trama de la muñeca a partir del vector de articulación. Si deseamos calcular sólo la posición cartesiana y no los ángulos de articulación es posible derivar ecuaciones que correlacionen directamente al espacio de actuador con el espacio cartesiano. Estas ecuaciones son un poco más simples en cuanto al aspecto computacional que el método de dos pasos. (Vea el ejercicio 3.10).

3.8 TRAMAS CON NOMBRES ESTÁNDAR

Por cuestión de convención será útil si asignamos nombres y ubicaciones específicas a ciertas tramas "estándar" asociadas con un robot y su espacio de trabajo. La figura 3.27 muestra una situación común en la que un robot ha sujetado algún tipo de herramienta y está por colocar la punta de la herramienta en una ubicación definida por el usuario. Se hace referencia a las cinco tramas indicadas en la figura 3.27 con tanta frecuencia que definiremos nombres para ellas. El nombramiento y uso subsecuente de estas cinco tramas en un sistema de programación y control de robots facilita crear habilidades generales en una manera fácilmente comprensible. Todos los movimientos del robot se describirán en términos de estas tramas.

A continuación veremos unas breves definiciones de las tramas que se muestran en la figura 3.27.

Trama base, {B}

$\{B\}$ está ubicada en la base del manipulador. Es simplemente otro nombre para la trama $\{0\}$. Está sujeta a una parte inmóvil del robot, algunas veces llamada vínculo 0.

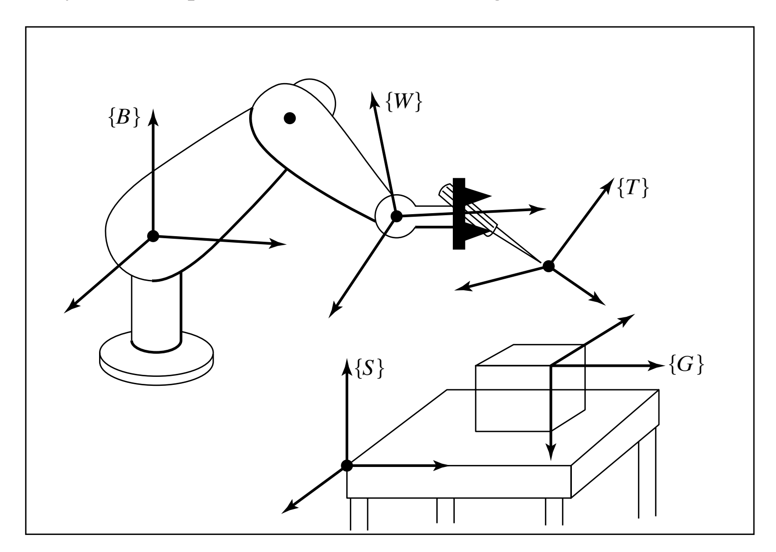


FIGURA 3.27: Tramas estándar.

Trama estación, {S}

{S} está ubicada en una posición relacionada con la tarea en cuestión. En la figura 3.28 se encuentra en la esquina de una mesa sobre la cual va a trabajar el robot. En lo que respecta al usuario de este sistema de robot, {S} es la trama universal y todas las acciones del robot se realizan en forma relativa a la misma. Algunas veces se le llama trama de tarea, trama mundial o trama universal. La trama estación siempre se especifica con respecto a la trama base, que viene siendo ${}^{B}_{S}T$.

Trama muñeca, {W}

{W} está sujeta al último vínculo del manipulador. Es otro nombre para la trama {N}, la trama de vínculo unida al último vínculo del robot. Es muy común que {W} tenga su origen fijo en un punto llamado la muñeca del manipulador, y {W} se mueve con el último vínculo del manipulador. Se define de manera relativa a la trama base, es decir, $\{W\} = {}^{B}_{W}T = {}^{0}_{N}T$.

Trama herramienta, {T}

{T} está sujeta al extremo de cualquier herramienta que el robot esté sujetando. Cuando la mano está vacía, {T} generalmente se ubica con su origen entre las puntas de los dedos del robot. La trama herramienta siempre se especifica con respecto a la trama muñeca. En la figura 3.28 la trama herramienta se define con su origen en la punta de un perno que el robot está sujetando.

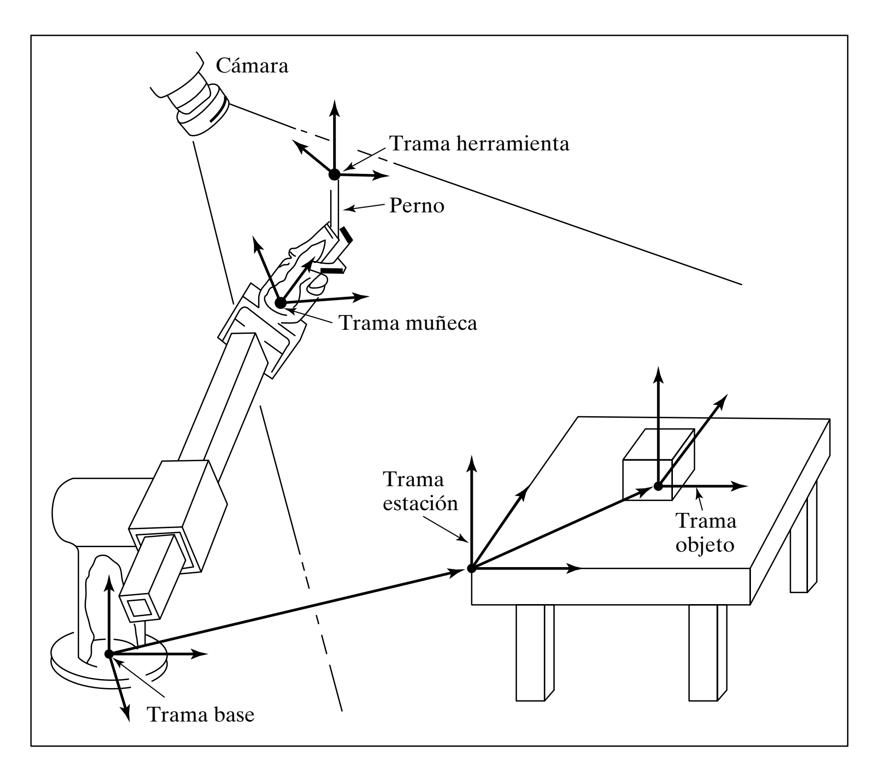


FIGURA 3.28: Ejemplo de la asignación de tramas estándar.

Trama objetivo, {G}

$\{G\}$ es una descripción de la ubicación a la cual el robot va a mover la herramienta. Específicamente esto significa que al final del movimiento la trama herramienta debe moverse para que coincida con la trama objetivo. $\{G\}$ siempre se especifica en relación con la trama estación. En la figura 3.28 el objetivo se encuentra en un hoyo en el que deseamos insertar un perno.

Todos los movimientos de los robots pueden describirse en términos de estas tramas sin perder generalidad. Su uso nos proporciona un lenguaje estándar para hablar sobre las tareas de los robots.

3.9 ¿EN DÓNDE ESTÁ LA HERRAMIENTA?

Una de las primeras habilidades que debe tener un robot es poder calcular la posición y orientación de la herramienta que está sujetando (o de su mano vacía) con respecto a un sistema de coordenadas conveniente. Esto es, deseamos calcular el valor de la trama herramienta $\{T\}$ relativa a la trama estación $\{S\}$. Una vez que se haya calculado ${}^{B}_{W}T$ mediante las ecuaciones cinemáticas podemos usar transformadas cartesianas, como se estudió en el capítulo 2, para calcular $\{T\}$ con respecto a $\{S\}$. Al resolver una simple ecuación de transformadas obtenemos

$$ {}^{S}_{T}T = {}^{B}_{S}T^{-1}\ {}^{B}_{W}T\ {}^{W}_{T}T. \tag{3.18}$$

La ecuación (3.18) implementa lo que se conoce como la función **DONDE** en algunos sistemas de robots. Calcula en "dónde" se encuentra el brazo. Para la situación en la figura 3.28, la salida de **DONDE** sería la posición y orientación del perno, en relación a la cubierta de la mesa.

Se puede considerar que la ecuación (3.18) *generaliza* la cinemática. ${}^{S}_{T}T$ calcula la cinemática debida a la geometría de las articulaciones, junto con una transformada general (que podría considerarse como un vínculo fijo) en el extremo de la base (${}^{B}_{S}T$) y otra en el efector final (${}^{W}_{T}T$). Estas transformadas adicionales nos permiten incluir herramientas con desplazamientos y torceduras, además de operar con respecto a una trama estación arbitraria.

3.10 CONSIDERACIONES COMPUTACIONALES

En muchos sistemas manipuladores prácticos hay que tener en cuenta el tiempo requerido para realizar cálculos cinemáticos. En esta sección hablaremos brevemente sobre varias cuestiones relacionadas con el cálculo de la cinemática de los manipuladores, como se ejemplifica mediante la ecuación (3.14) para el caso del PUMA 560.

Una opción a elegir es el uso de la representación fija o de punto flotante para las cantidades involucradas. Muchas implementaciones usan punto flotante para facilitar el desarrollo del software, ya que el programador no tiene que preocuparse por las operaciones de escala que capturan las magnitudes relativas de las variables. No obstante, cuando la velocidad es crucial es muy factible usar la representación de punto fijo, ya que las variables no tienen un rango dinámico extenso, y estos rangos son bastante conocidos. Una estimación aproximada del número de bits necesarios en la representación de punto fijo parece indicar que 24 son suficientes [4].

Al factorizar ecuaciones tales como la (3.14) es posible reducir el número de multiplicaciones y sumas, con la desventaja de crear variables locales (lo que generalmente es aceptable). La idea es evitar términos comunes de cómputo una y otra vez a lo largo del cálculo. Se ha hecho cierta aplicación de factorización automática asistida por computadora para dichas ecuaciones [5].

El mayor problema de los cálculos cinemáticos es a menudo calcular las funciones trascendentales (seno y coseno). Cuando estas funciones están disponibles como parte de una biblioteca estándar, a menudo se hacen a partir de una expansión de series, con el riesgo de ocupar muchas multiplicaciones. Muchos sistemas de manipulación (con el riesgo de requerir cierta cantidad de memoria) emplean implementaciones de búsqueda en tabla de las funciones trascendentales. Dependiendo del esquema, esto reduce la cantidad de tiempo requerido para calcular un seno o coseno a dos o tres multiplicaciones o menos [6].

El cálculo cinemático, como el mostrado en la ecuación (3.14), es redundante, ya que fue necesario calcular nueve cantidades para representar la orientación. Una manera de reducir el trabajo es determinar sólo dos columnas de la matriz de rotación y luego determinar un producto cruz (el cual requiere solamente seis multiplicaciones y tres sumas) para determinar la tercera columna. Obviamente, se eligen las dos columnas menos complicadas para efectuar las operaciones matemáticas.

BIBLIOGRAFÍA

[1] J. Denavit y R.S. Hartenberg, “A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices”, *Journal of Applied Mechanics*, pp. 215-221, junio de 1955.

[3] J. Lenarčič, “Kinematics”, en *The International Encyclopedia of Robotics*, R. Dorf y S. Nof, Editores, John C. Wiley and Sons, Nueva York, 1988.

[3] J. Colson y N.D. Perreira, “Kinematic Arrangements Used in Industrial Robots”, 1*3th Industrial Robots Conference Proceedings*, abril de 1983.

[4] T. Turner, J. Craig y W. Gruver, “A Microprocessor Architecture for Advanced Robot Control”, 14th ISIR, Estocolmo, Suecia, octubre de 1984.

[5] W. Schiehlen, “Computer Generation of Equations of Motion”, en *Computer Aided Analysis and Optimization of Mechanical System Dynamics*, E.J. Haug, Editor, Springer-Verlag, Berlín y Nueva York, 1984.

[6] C. Ruoff, “Fast Trigonometric Functions for Robot Control”, *Robotics Age*, noviembre de 1981.

EJERCICIOS

3.1 [15] Calcule la cinemática del brazo planar del ejemplo 3.3.

3.3 [37] Imagine un brazo como el PUMA 560, excepto que se remplace la articulación 3 con una articulación prismática. Suponga que la articulación prismática se desliza a lo largo de la dirección de $\hat{X}_1$ en la figura 3.18; no obstante, sigue habiendo un desplazamiento equivalente a d_3 que se debe tener en cuenta. Haga todas las suposiciones adicionales necesarias. Derive las ecuaciones cinemáticas.

3.3 [25] El brazo con tres grados de libertad que se muestra en la figura 3.29 es como el del ejemplo 3.3, excepto que el eje de la articulación 1 no es paralelo a los otros dos. En su lugar hay una torsión de 90 grados de magnitud entre los ejes 1 y 2. Derive los parámetros de vínculo y las ecuaciones cinemáticas para ${}^{B}_{W}T$. Observe que no es necesario definir l_3.

3.4 [22] En el brazo con tres grados de libertad que se muestra en la figura 3.30, las articulaciones 1 y 2 son perpendiculares y las articulaciones 2 y 3 son paralelas. Según lo ilustrado, todas las articulaciones están en su ubicación cero. Observe que se indica el sentido positivo del ángulo de articulación. Asigne las tramas de vínculo de la {0} a la {3} para este brazo; esto es, haga un bosquejo del brazo mostrando la articulación de las tramas. Después derive las matrices de transformación ${}^{0}_{1}T$, ${}^{1}_{2}T$ y ${}^{2}_{3}T$.

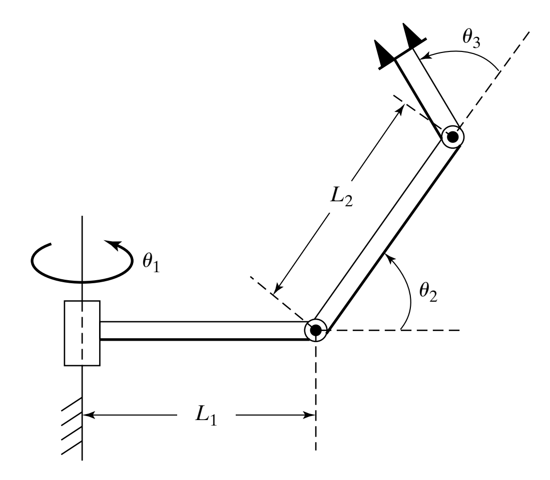


FIGURA 3.29: El brazo $3R$ no planar (ejercicio 3.3).

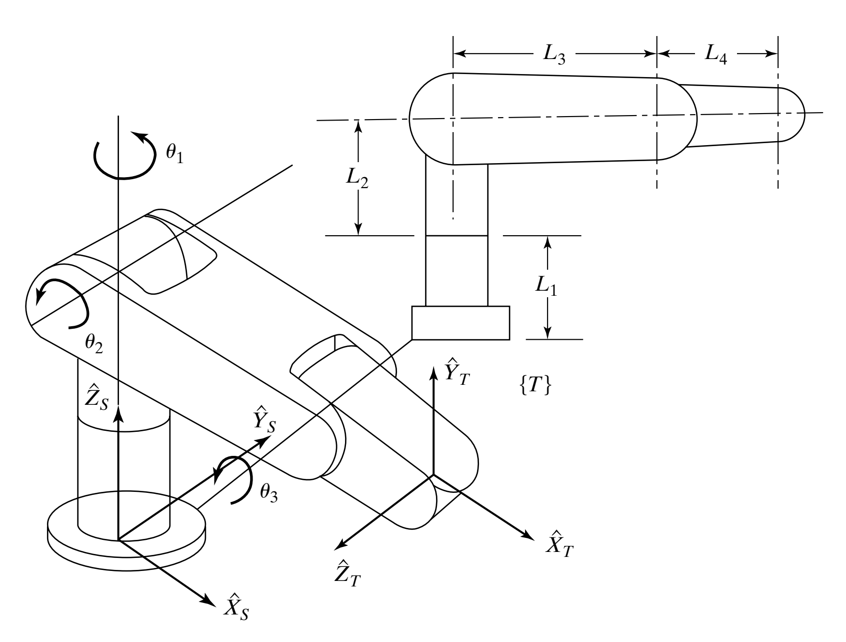


FIGURA 3.30: Dos vistas de un manipulador $3R$ (ejercicio 3.4).

3.5 [26] Escriba una subrutina para calcular la cinemática de un PUMA 560. Codifique teniendo en cuenta la velocidad, tratando de minimizar el número de multiplicaciones lo más que sea posible. Use el siguiente encabezado de procedimiento (o su equivalente en C):

```
Procedure CIN (VAR theta: vec6; VAR wrelb: trama);
```

Cuente una evaluación de seno o coseno como 5 multiplicaciones. Considere las sumas como 0.333 multiplicaciones y las instrucciones de asignación como 0.2 multipli-

caciones. Considere que el cálculo de una raíz cuadrada toma 4 multiplicaciones. ¿Cuántas multiplicaciones necesita?

3.6 [20] Escriba una subrutina para calcular la cinemática del brazo cilíndrico del ejemplo 3.4. Utilice el siguiente encabezado de procedimiento (o su equivalente en C):

```
Procedure CIN(VAR varartic: vec3; VAR wrelb: trama);
```

Cuente una evaluación de seno o coseno como 5 multiplicaciones. Considere las sumas como 0.333 multiplicaciones y las instrucciones de asignación como 0.2 multiplicaciones. Cuente un cálculo de raíz cuadrada como 4 multiplicaciones. ¿Cuántas multiplicaciones necesita?

3.7 [22] Escriba una subrutina para calcular la cinemática del brazo del ejercicio 3.3 Utilice el siguiente encabezado de procedimiento (o su equivalente en C):

```
Procedure CIN(VAR theta: vec3; VAR wrelb: trama);
```

Cuente una evaluación de seno o coseno como 5 multiplicaciones. Cuente las sumas como 0.333 multiplicaciones y las instrucciones de asignación como 0.2 multiplicaciones. Considere que el cálculo de una raíz cuadrada toma 4 multiplicaciones. ¿Cuántas multiplicaciones necesita?

3.8 [13] En la figura 3.31 no se conoce la ubicación exacta de la herramienta ${}^{W}_{T}T$. Usando el control de fuerza, el robot siente los alrededores con la punta de la herramienta hasta que la inserta en el receptáculo (u objetivo) en la ubicación ${}^{S}_{G}T$. Una vez en esta configuración de "calibración" (en la que $\{G\}$ y $\{T\}$ son coincidentes), la posición del robot ${}^{B}_{W}T$ se averigua tomando lectura de los sensores de los ángulos de articulación y calculando la cinemática. Suponiendo que se conozcan ${}^{B}_{S}T$ y ${}^{S}_{G}T$, proporcione la ecuación de transformadas para calcular la trama herramienta desconocida ${}^{W}_{T}T$.

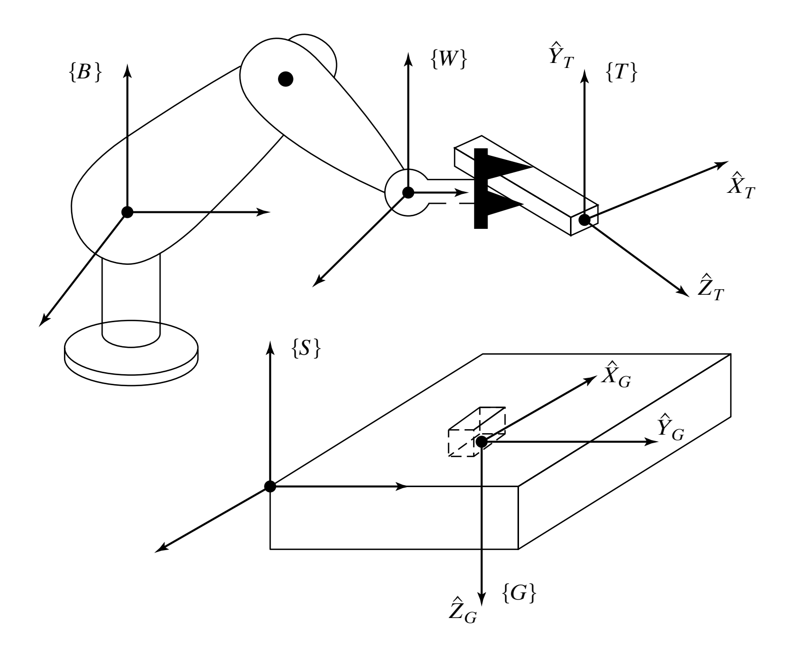


FIGURA 3.31: Determinación de la trama herramienta (ejercicio 3.8).

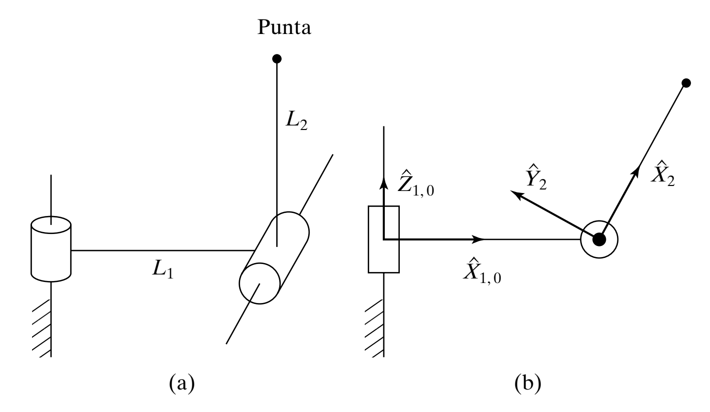


FIGURA 3.32: Brazo de dos vínculos con asignaciones de tramas (ejercicio 3.9).

3.9 [11] Para el manipulador de dos vínculos que se muestra en la figura 3.32(a) se construyeron las matrices de transformación de vínculos ${}^0_1T$ y ${}^1_2T$. Su producto es:

$$ {}^0_2T = \begin{bmatrix} c\theta_1 c\theta_2 & -c\theta_1 s\theta_2 & s\theta_1 & l_1 c\theta_1 \\ s\theta_1 c\theta_2 & -s\theta_1 s\theta_2 & -c\theta_1 & l_1 s\theta_1 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

Las asignaciones de trama de vínculo que se utilizan se indican en la figura 3.32(b). Observe que la trama {0} es coincidente con la trama {1} cuando $\theta_1 = 0$. La longitud del segundo vínculo es l_2. Encuentre una expresión para el vector ${}^0P_{punta}$, el cual posiciona la punta del brazo con respecto a la trama {0}.

3.10 [39] Derive ecuaciones cinemáticas para el robot Yasukawa Motoman (vea la sección 3.7) que calculan la posición y la orientación de la trama de la muñeca directamente de los valores de los actuadores, en vez de calcular primero los ángulos de articulación. Es posible encontrar una solución que requiera solamente de 33 multiplicaciones, dos raíces cuadradas y seis evaluaciones de senos o cosenos.

3.11 [17] La figura 3.33 muestra el diagrama esquemático de una muñeca que tiene tres ejes perpendiculares que no son ortogonales. Asigne tramas de vínculos a esta muñeca (como si fuera un manipulador de tres grados de libertad o 3-DOF) y proporcione los parámetros de vínculos.

3.12 [08] ¿Puede una transformación de cuerpo rígido arbitraria expresarse con cuatro parámetros (a, α, d, θ) en la forma de la ecuación (3.6)?

3.13 [15] Muestre la articulación de las tramas de vínculos para el manipulador de cinco grados de libertad o 5-DOF que se muestra esquemáticamente en la figura 3.34.

3.14 [20] Como se dijo anteriormente, la posición relativa de dos líneas cualesquiera en el espacio puede proporcionarse mediante dos parámetros, a y α, en donde a es la longitud de la perpendicular común que une las dos líneas y α es el ángulo formado por los dos ejes cuando se proyectan en un plano que sea normal a la perpendicular común. Dada una línea que se define como la que pasa a través del punto p, con la dirección del vector unitario $\hat{m}$ y una segunda línea que pasa a través del punto q con la dirección del vector unitario $\hat{n}$, escriba expresiones para a y α.

3.15 [15] Muestre la articulación de tramas de vínculos para el manipulador 3-DOF que se muestra esquemáticamente en la figura 3.35.

3.16 [15] Asigne tramas de vínculos al robot planar *RPR* que se muestra en la figura 3.36 y proporcione los parámetros de las articulaciones.

3.17 [15] Muestre la articulación de tramas de vínculos en el robot de tres vínculos que se muestra en la figura 3.37.

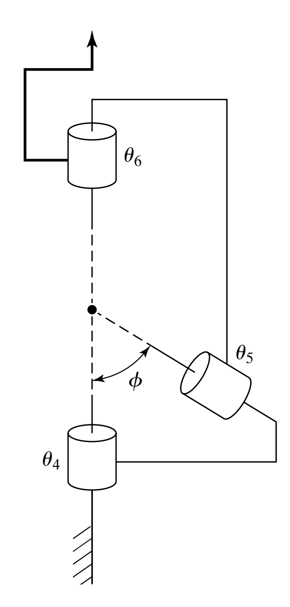


FIGURA 3.33: Robot 3*R* con eje no ortogonal (ejercicio 3.11).

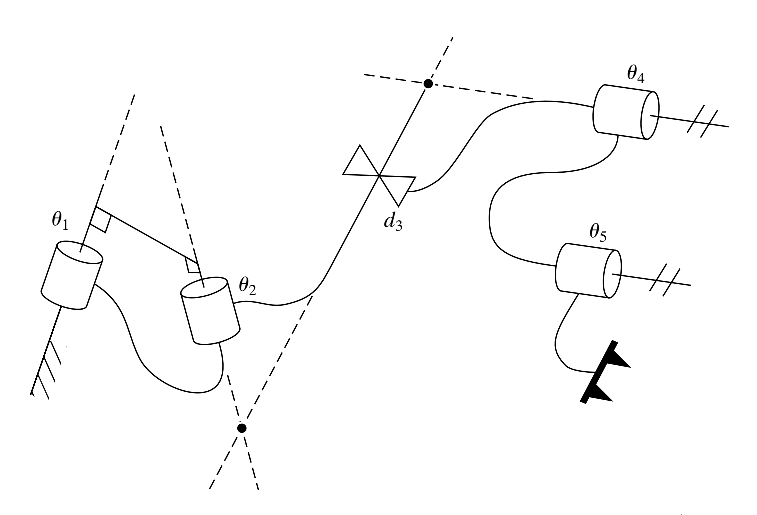


FIGURA 3.34: Diagrama esquemático de un manipulador 2*RP*2*R* (ejercicio 3.13).

3.18 [15] Muestre la articulación de tramas de vínculos en el robot de tres vínculos que se muestra en la figura 3.38.

3.19 [15] Muestre la articulación de tramas de vínculos en el robot de tres vínculos que se muestra en la figura 3.39.

3.20 [15] Muestre la articulación de tramas de vínculos en el robot de tres vínculos que se muestra en la figura 3.40.

3.21 [15] Muestre la articulación de tramas de vínculos en el robot de tres vínculos que se muestra en la figura 3.41.

3.22 [15] Muestre la articulación de tramas de vínculos en el robot de tres vínculos que se muestra en la figura 3.42. Dadas sus asignaciones de tramas, ¿cuáles son los signos de d_2, d_3 y a_2?

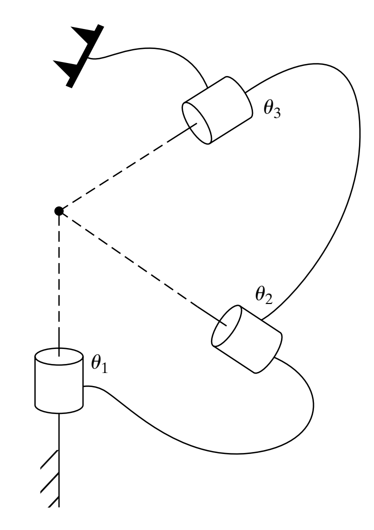


FIGURA 3.35: Diagrama esquemático de un manipulador $3R$ (ejercicio 3.15).

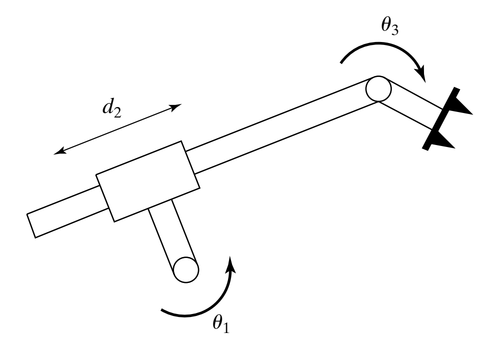


FIGURA 3.36: Robot planar RPR (ejercicio 3.16).

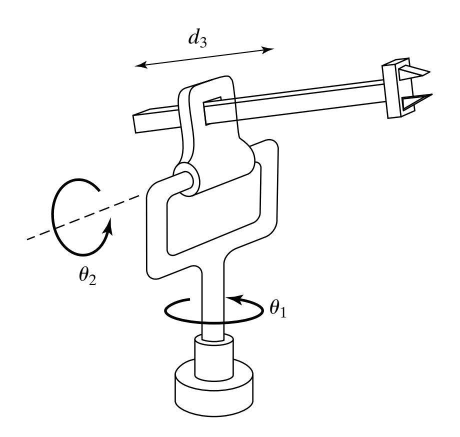


FIGURA 3.37: Manipulador RRP de tres vínculos (ejercicio 3.17).

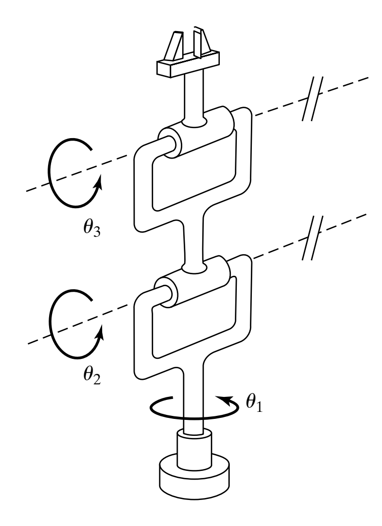


FIGURA 3.38: Manipulador *RRR* de tres vínculos (ejercicio 3.18).

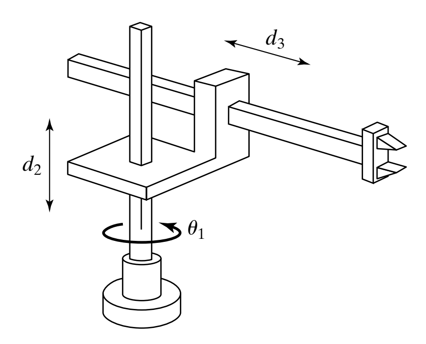


FIGURA 3.39: Manipulador *RPP* de tres vínculos (ejercicio 3.19).

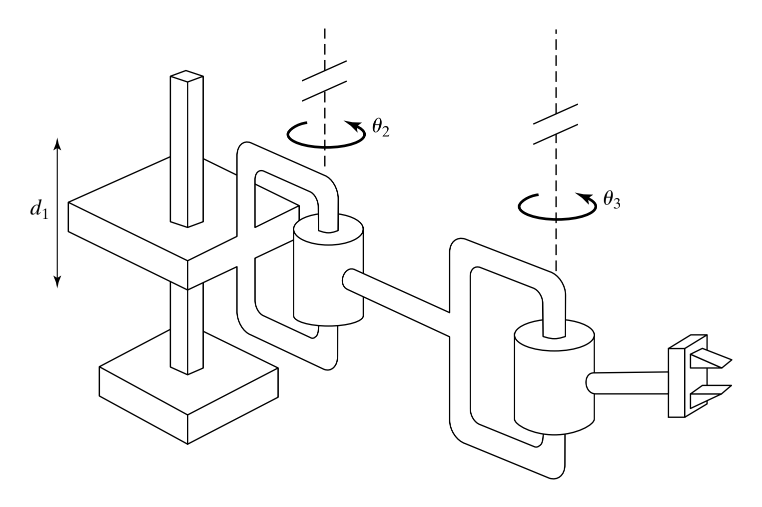


FIGURA 3.40: Manipulador *PRR* de tres vínculos (ejercicio 3.20).

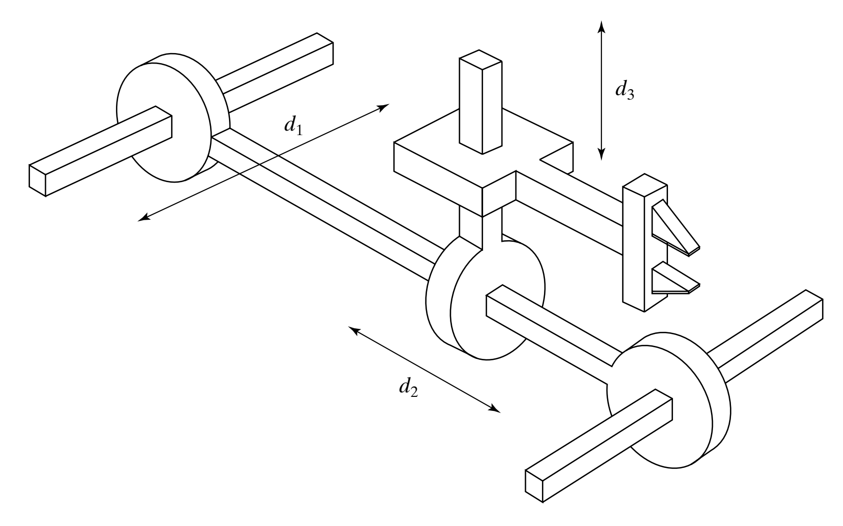


FIGURA 3.41: Manipulador *PPP* de tres vínculos (ejercicio 3.21).

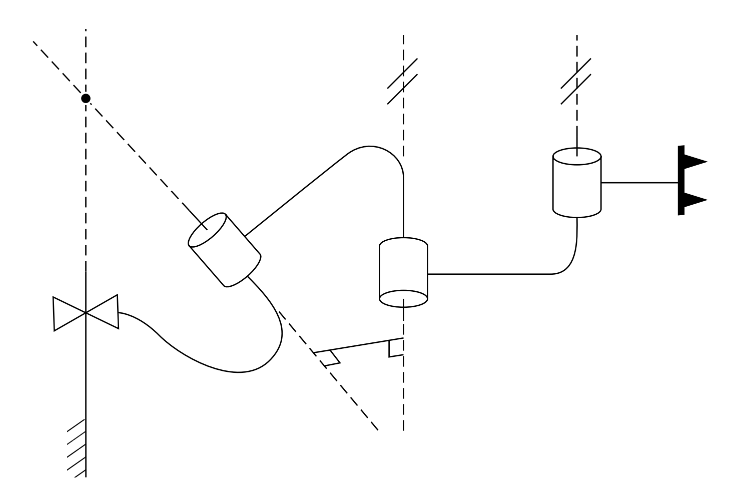


FIGURA 3.42: Diagrama esquemático de un manipulador *P3R* (ejercicio 3.22).

EJERCICIO DE PROGRAMACIÓN (PARTE 3)

1. Escriba una subrutina para calcular la cinemática del robot *3R* planar del ejemplo 3.3; es decir, una rutina con los valores de los ángulos de articulación como entrada, y una trama (la de muñeca relativa a la trama base) como salida. Use el siguiente encabezado de procedimiento (o su equivalente en C):

```
Procedure CIN(VAR theta: vec3; VAR wrelb: trama);
```

en donde “wrelb” es la trama de muñeca relativa a la trama base, ${}^{B}_{W}T$. El tipo “trama” consiste de una matriz de rotación de 2×2 y un vector posición de 2×1. Si lo desea, puede representar la trama con una transformada homogénea de 3×3 en la que la tercera fila es [0 0 1]. (Los datos del manipulador son $l_1 = l_2 = 0.5$ metros).

2. Escriba una rutina que calcule en dónde está la herramienta respecto a la trama estación. La entrada a la rutina debe ser un vector de ángulos de articulación:

```
Procedure DONDE(VAR theta: vec3; VAR trels: trama);
```

Obviamente, DONDE debe hacer uso de descripciones de la trama herramienta y la trama base del robot para poder calcular la ubicación de la herramienta, relativa a la trama estación. Los valores de ${}^{W}_{T}T$ y ${}^{S}_{B}T$ deben almacenarse en la memoria global (o, como segunda opción, pueden pasarse como argumentos en la rutina DONDE).

3. El usuario define una trama herramienta y una trama estación para cierta tarea, como se muestra a continuación:

$$ {}^{W}_{T}T = [x \ y \ \theta] = [0.1 \ 0.2 \ 30.0], $$

$$ {}^{B}_{S}T = [x \ y \ \theta] = [-0.1 \ 0.3 \ 0.0]. $$

Calcule la posición y la orientación de la herramienta respecto a la trama estación para las siguientes tres configuraciones del brazo (en unidades de grados):

$$ [\theta_1 \ \theta_2 \ \theta_3] = [0.0 \ 90.0 \ -90.0], $$

$$ [\theta_1 \ \theta_2 \ \theta_3] = [-23.6 \ -30.3 \ 48.0], $$

$$ [\theta_1 \ \theta_2 \ \theta_3] = [130.0 \ 40.0 \ 12.0]. $$

EJERCICIO MATLAB 3

Este ejercicio se enfoca en los parámetros DH y en el planteamiento directo de la transformación cinemática (posición y orientación) para el robot de 3 grados de libertad, o 3-DOF, 3R planar (de las figuras 3.6 y 3.7). Se proporcionan los siguientes parámetros de longitud fija: $L_1 = 4$, $L_2 = 3$ y $L_3 = 2$ (m).

a) Evalúe los parámetros DH. Puede comprobar sus resultados con la figura 3.8.

b) Evalúe las matrices de transformación homogéneas adyacentes ${}^{i-1}_{i}T$, $i = 1, 2, 3$, las cuales son funciones de las variables de ángulo de articulación θ_i, $i = 1, 2, 3$. Además, evalúe la constante ${}^{3}_{H}T$ por inspección: el origen de $\{H\}$ está en el centro de los dedos de la herramienta de sujeción y la orientación de $\{H\}$ es siempre la misma que la orientación de $\{3\}$.

c) Use el software Symbolic MATLAB para determinar la solución cinemática de planteamiento directo ${}^{0}_{3}T$ y ${}^{0}_{H}T$ simbólicamente (como una función de θ_i). Abrevie su respuesta usando $s_i = \text{sen}(\theta_i)$, $c_i = \cos(\theta_i)$, y así sucesivamente. Además hay una simplificación de $(\theta_1 + \theta_2 + \theta_3)$ mediante el uso de fórmulas de la suma de ángulos, esto debido a los ejes Z_i paralelos. Calcule los resultados de cinemática de planteamiento directo (tanto ${}^{0}_{3}T$ como ${}^{0}_{H}T$) mediante MATLAB para los siguientes casos de entrada:

i) $\Theta = \{\theta_1 \ \theta_2 \ \theta_3\}^T = \{0 \ 0 \ 0\}^T$.

ii) $\Theta = \{10^\circ \ 20^\circ \ 30^\circ\}^T$.

iii) $\Theta = \{90^\circ \ 90^\circ \ 90^\circ\}^T$.

Para los tres casos, compruebe sus resultados con un bosquejo de la configuración del manipulador y determinando la transformación cinemática de planteamiento directo por inspección. (Piense en la definición de ${}^{0}_{H}T$ en términos de una matriz de rotación y un vector posición.) Incluya las tramas $\{H\}$, $\{3\}$ y $\{0\}$ en sus bosquejos.

d) Compruebe todos sus resultados mediante la Caja de Herramientas de Robótica MATLAB de Corke. Pruebe las funciones *link*(), *robot*() y *fkine*().

CAPÍTULO 4

Cinemática inversa de manipuladores

4.1 INTRODUCCIÓN

En el capítulo 3 consideramos el problema de calcular la posición y la orientación de la herramienta respecto a la estación de trabajo del usuario cuando se proporcionan los ángulos de articulación del manipulador. En este capítulo investigaremos el problema converso, más difícil: dada la posición y orientación deseadas de la herramienta respecto a la estación, ¿cómo calculamos el conjunto de ángulos de articulación que logren este resultado deseado? Mientras que el capítulo 3 trató sobre la **cinemática directa** de los manipuladores, aquí el énfasis será en la **cinemática inversa** de los mismos.

La manera de resolver el problema de encontrar los ángulos de articulación requeridos para colocar la trama herramienta $\{T\}$ relativa a la trama estación $\{S\}$ se divide en dos partes. Primero se realizan transformaciones de trama para encontrar la trama de la muñeca $\{W\}$ relativa a la trama base $\{B\}$, y después se utiliza cinemática inversa para resolver los ángulos de articulación.

4.2 SOLUBILIDAD

El problema de resolver las ecuaciones cinemáticas de un manipulador es no lineal. Dado el valor numérico de ${}^0_N T$, tratamos de encontrar valores de $\theta_1, \theta_2, \ldots, \theta_n$. Considere las ecuaciones en (3.14). En el caso del manipulador PUMA 560, la declaración precisa de nuestro problema actual sería la siguiente: dada ${}^0_6 T$ como 16 valores numéricos (cuatro de los cuales son triviales), resuelva (3.14) para los seis ángulos de articulación θ_1 al θ_6.

Para el caso de un brazo con seis grados de libertad (como el que corresponde a las ecuaciones en (3.14)), tenemos 12 ecuaciones y seis variables desconocidas. No obstante, entre las nueve ecuaciones que surgen de la porción de la matriz de rotación de ${}^0_6 T$, sólo tres son independientes. Éstas, agregadas a las tres ecuaciones de la porción

del vector de posición de ${}^{0}_{6}T$, proporcionan seis ecuaciones con seis variables desconocidas. Estas ecuaciones son no lineales y trascendentales, por lo cual pueden ser bastante difíciles de resolver. Las ecuaciones de (3.14) son las de un robot que tiene parámetros de vínculo muy simples; muchas de las α_i son 0 o ± 90 grados. Muchos desplazamientos de vínculo y longitudes son cero. Es fácil imaginar que, para el caso del mecanismo general con seis grados de libertad (con todos los parámetros de vínculo distintos de cero), las ecuaciones cinemáticas serían mucho más complejas que las de (3.14). Al igual que con cualquier conjunto no lineal de ecuaciones, debemos preocuparnos por la existencia de soluciones, por los casos con soluciones múltiples y por el método de solución.

Existencia de soluciones

La pregunta de si existe alguna solución hace que surja otra sobre el **espacio de trabajo** del manipulador. En general, el espacio de trabajo es el volumen de espacio que puede ser alcanzado por el efector final del manipulador. Para que exista una solución el punto de destino debe estar dentro del espacio de trabajo. Algunas veces es útil considerar dos definiciones sobre el espacio de trabajo: **espacio de trabajo diestro** es el volumen de espacio que puede alcanzar el efector final del robot con todas las orientaciones; esto es, en cada punto del espacio de trabajo diestro, el efector final puede orientarse arbitrariamente. El **espacio de trabajo alcanzable** es el volumen de espacio que puede alcanzar el robot en por lo menos una orientación. Evidentemente, el espacio de trabajo diestro es un subconjunto del espacio de trabajo alcanzable.

Considere el espacio de trabajo del manipulador de dos vínculos en la figura 4.1. Si $l_1 = l_2$, entonces el espacio de trabajo alcanzable consiste en un disco de radio $2l_1$. El espacio de trabajo diestro consiste de un solo punto, el origen. Si $l_1 \neq l_2$, entonces no hay espacio de trabajo diestro y el espacio de trabajo alcanzable se convierte en un anillo con un radio exterior de $l_1 + l_2$ y un radio interior de $|l_1 - l_2|$. Dentro del espacio de trabajo alcanzable hay dos posibles orientaciones del efector final. En los límites del espacio de trabajo sólo hay una orientación posible.

Estas consideraciones de espacio de trabajo para el manipulador de dos vínculos suponen que todas las articulaciones pueden girar 360 grados. Esto raras veces es cierto para los mecanismos reales. Cuando los límites de las articulaciones son un subconjunto de los 360 grados completos, es obvio que el espacio de trabajo se reduce en forma correspondiente, ya sea en extensión o en el número de orientaciones posibles que pue-

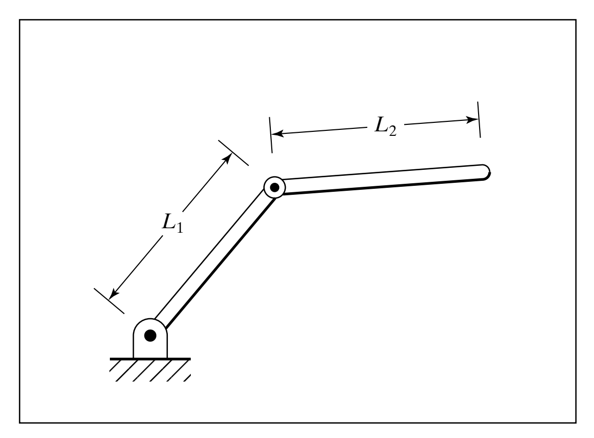


FIGURA 4.1: Manipulador de dos vínculos con longitudes de vínculo l_1 y l_2.

dan obtenerse. Por ejemplo, si el brazo de la figura 4.1 tiene un movimiento de 360 grados para θ_1, pero sólo $0 \leq \theta_2 \leq 180°$, entonces el espacio de trabajo alcanzable tiene la misma extensión pero sólo puede obtenerse una orientación en cada punto.

Cuando un manipulador tiene menos de seis grados de libertad, no puede alcanzar posiciones y orientaciones para un destino arbitrario en espacio 3D. Evidentemente el manipulador planar de la figura 4.1 no puede salirse del plano, por lo que cualquier punto de destino con un valor de coordenada Z distinto de cero puede rechazarse rápidamente como inalcanzable. En muchas situaciones realistas se emplean manipuladores con cuatro o cinco grados de libertad que operan fuera de un plano, pero que evidentemente no pueden alcanzar destinos arbitrarios. Cada uno de esos manipuladores debe estudiarse para comprender su espacio de trabajo. En general, el espacio de trabajo de dicho robot es un subconjunto de un subespacio que puede asociarse con cualquier robot en particular. Dada una especificación de trama de destino general, surge un problema interesante en relación con los manipuladores que tienen menos de seis grados de libertad: ¿cuál es la trama de destino más cercana que puede obtenerse?

El espacio de trabajo también depende de la transformación de la trama herramienta, ya que generalmente es la punta de la herramienta sobre la que se discute cuando hablamos de puntos alcanzables en el espacio. En general la transformación de la herramienta se realiza de manera independiente de las cinemáticas directa e inversa del manipulador, por lo que a menudo tenemos que considerar el espacio de trabajo de la trama de la muñeca, $\{W\}$. Para un efector final dado se define una trama herramienta $\{T\}$; dada una trama de destino $\{G\}$ se calcula la trama $\{W\}$ correspondiente y después preguntamos: ¿esta posición y orientación deseadas de $\{W\}$ se encuentran en el espacio de trabajo? De esta manera, el espacio de trabajo en el que debemos concentrarnos (en sentido computacional) es distinto del que imagina el usuario, que está concentrado en el espacio de trabajo del efector final (la trama $\{T\}$).

Si la posición y orientación deseadas de la trama de la muñeca se encuentran en el espacio de trabajo, entonces existe por lo menos una solución.

Múltiples soluciones

Otro posible problema que nos encontramos al resolver ecuaciones cinemáticas es el de múltiples soluciones. Un brazo planar con tres articulaciones angulares tiene un espacio de trabajo diestro en el plano extenso (dadas unas "buenas" longitudes de vínculo y rangos de articulación extensos), ya que cualquier posición en el interior de su espacio de trabajo puede alcanzarse con cualquier orientación. La figura 4.2 muestra un

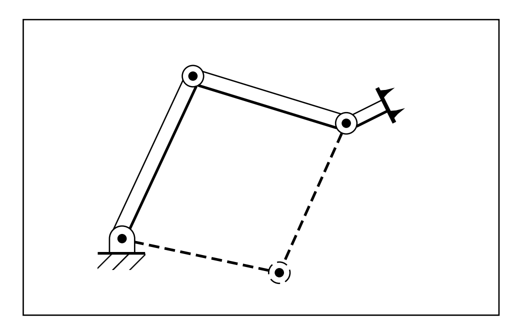


FIGURA 4.2: Manipulador de tres vínculos. Las líneas punteadas indican una segunda solución.

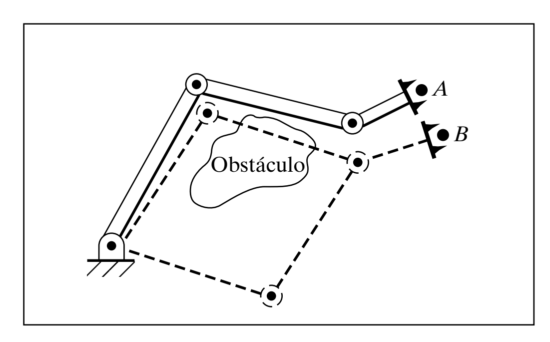


FIGURA 4.3: Una de las dos posibles soluciones para llegar al punto B produce una colisión.

brazo planar de tres vínculos con su efector final en cierta posición y orientación. Las líneas punteadas indican una segunda configuración posible en la que puede lograrse la misma posición y orientación del efector final.

El hecho de que un manipulador tenga múltiples soluciones puede ocasionar problemas, ya que el sistema tiene que ser capaz de elegir una. Los criterios sobre los que debe basarse una decisión varían, pero una elección muy razonable sería la solución *más cercana*. Por ejemplo, si el manipulador se encuentra en el punto A, como en la figura 4.3, y deseamos moverlo al punto B, una buena elección sería la solución que minimiza la cantidad que se requiere mover cada articulación. Por lo tanto, en la ausencia del obstáculo se elegiría la configuración superior marcada con líneas punteadas en la figura 4.3. Esto sugiere que un argumento de entrada para nuestro procedimiento cinemático inverso podría ser la posición actual del manipulador. De esta manera, si hay una opción, nuestro algoritmo puede elegir la solución más cercana en espacio de articulación. No obstante, la noción de “cercano” podría definirse de varias formas. Por ejemplo, los robots comunes podrían tener tres vínculos extensos seguidos de tres vínculos de orientación más pequeños cerca del efector final. En este caso podría aplicarse una ponderación en el cálculo de cuál solución es “más cercana”, de manera que la selección favoreciera las articulaciones móviles más pequeñas, en vez de mover las articulaciones grandes, cuando exista una opción. La presencia de obstáculos podría forzar la elección de una solución “más lejana” en casos en los que la solución “cercana” produjera una colisión; entonces, en general necesitamos poder calcular todas las soluciones posibles. Por lo anterior, en la figura 4.3 la presencia del obstáculo implica que se va a utilizar la configuración marcada con las líneas punteadas inferiores para llegar al punto B.

El número de soluciones depende del número de articulaciones en el manipulador, pero también es una función de los parámetros de vínculo (α_i, a_i y d_i para un manipulador con articulación giratoria) y los rangos de movimiento permisibles de las articulaciones. Por ejemplo, el PUMA 560 puede llegar a ciertos destinos con ocho soluciones distintas. La figura 4.4 muestra cuatro soluciones; todas colocan la mano con la misma posición y orientación. Para cada solución descrita existe otra solución en la que las últimas tres articulaciones se cambian a una configuración alterna, de acuerdo con las siguientes fórmulas:

$$\begin{aligned} \theta_4' &= \theta_4 + 180°, \\ \theta_5' &= -\theta_5, \\ \theta_6' &= \theta_6 + 180°. \end{aligned} \tag{4.1}$$

Por lo tanto, en total puede haber ocho soluciones para un solo destino. Debido a los límites sobre los rangos de articulación algunas de estas soluciones podrían ser inaccesibles.

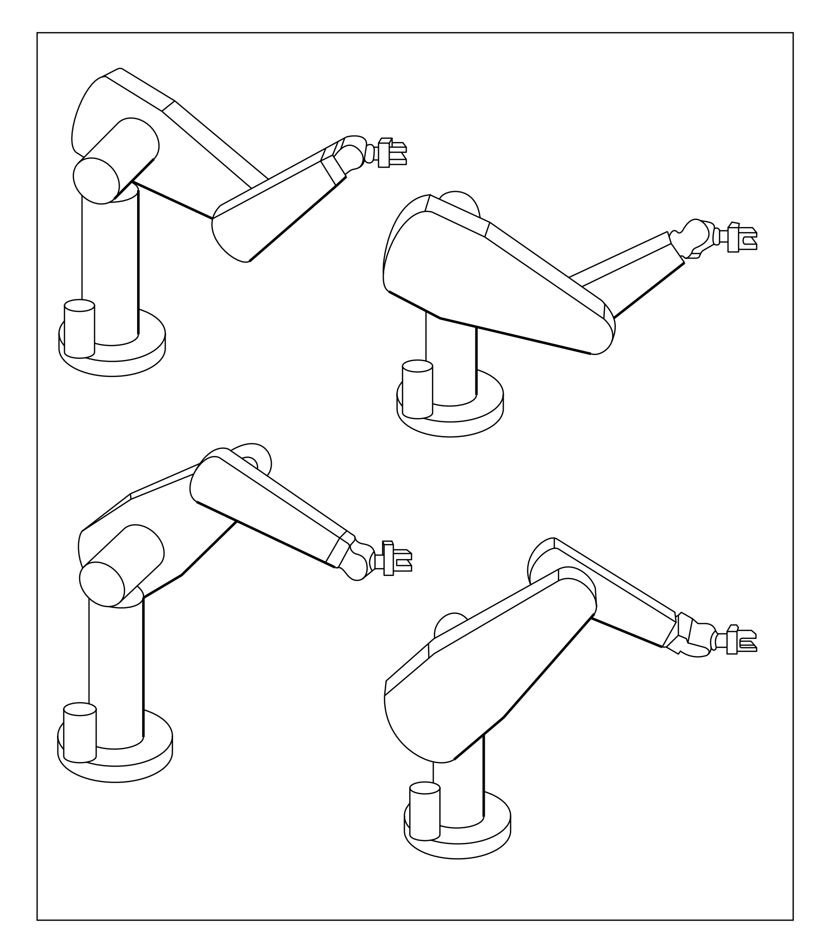


FIGURA 4.4: Cuatro soluciones para el PUMA 560.

En general, entre más parámetros de vínculo distintos de cero haya, más formas habrá de llegar a un cierto destino. Por ejemplo, considere un manipulador con seis articulaciones giratorias. La figura 4.5 muestra cómo se relaciona el máximo número de soluciones con el número de los parámetros de longitud de vínculo (a_i) que son cero. Entre más parámetros sean distintos de cero, mayor será el máximo número de soluciones. Para un manipulador general de seis grados de libertad con articulaciones giratorias, hay hasta 16 soluciones posibles [1,6].

Método de solución

A diferencia de las ecuaciones lineales, no hay algoritmos generales que puedan emplearse para resolver un conjunto de ecuaciones no lineales. Al considerar métodos de solución, es conveniente definir lo que constituye la "solución" de un manipulador dado.

Se considerará que un manipulador tiene solución si las variables de articulación pueden determinarse mediante un algoritmo que nos permita determinar *todos* los conjuntos de variables de articulación asociados con una posición y orientación dadas [2].

a_i	Número de soluciones
$a_1 = a_3 = a_5 = 0$	$\leqslant 4$
$a_3 = a_5 = 0$	$\leqslant 8$
$a_3 = 0$	$\leqslant 16$
Todas las $a_i \neq 0$	$\leqslant 16$

FIGURA 4.5: Comparación entre el número de soluciones y los valores de a_i distintos de cero.

El punto principal de esta definición es que, en el caso de múltiples soluciones, requerimos que sea posible calcular todas las soluciones. Por lo tanto, no consideramos que ciertos procedimientos iterativos numéricos resuelvan el manipulador en la medida en que no se garantiza que esos métodos encuentren todas las soluciones.

Dividiremos todas las estrategias de solución de manipuladores propuestas en dos clases amplias: **soluciones de forma cerrada (analíticas)** y **soluciones numéricas**. Debido a su naturaleza iterativa, las soluciones numéricas son generalmente mucho más lentas que la solución de forma cerrada correspondiente; de hecho, son tan lentas que para casi todos los fines no estaremos interesados en el enfoque numérico para la solución de problemas cinemáticos. La solución iterativa numérica para las ecuaciones cinemáticas es todo un campo de estudio en sí (vea [6,11,12]) y se encuentra más allá del alcance de este texto.

Restringiremos nuestra atención a los métodos de solución de forma cerrada. En este contexto, “forma cerrada” significa un método de solución basado en expresiones analíticas o en la solución de un polinomio de grado 4 o menor, de tal forma que basta con los cálculos no iterativos para llegar a una solución. Dentro de la clase de soluciones de forma cerrada, distinguimos dos métodos para obtener la solución: **algebraico** y **geométrico**. Estas distinciones son algo confusas: cualquier método geométrico a considerar se aplica por medio de expresiones algebraicas, por lo que los dos métodos son similares. Los métodos tal vez difieren sólo en el enfoque.

Un importante resultado reciente en cinemática es que, de acuerdo con nuestra definición de solubilidad, *todos los sistemas con articulaciones angulares y prismáticas que tengan un total de seis grados de libertad en una sola cadena en serie pueden resolverse.* No obstante, esta solución general es numérica. Sólo en casos especiales los robots con seis grados de libertad pueden resolverse analíticamente. Estos robots para los que existe una solución analítica (o de forma cerrada) se caracterizan ya sea por tener varios ejes de articulación perpendiculares o muchas α_i iguales a 0 o ± 90 grados. Generalmente el cálculo de soluciones numéricas consume mucho tiempo cuando se le compara con la evaluación de expresiones analíticas; por lo tanto, se considera muy importante diseñar un manipulador de manera que exista una solución de forma cerrada. Los diseñadores de manipuladores descubrieron esto muy pronto, por lo que ahora casi todos los manipuladores industriales están diseñados con la suficiente simplicidad como para que pueda desarrollarse una solución de forma cerrada.

Una condición suficiente para que un manipulador con seis articulaciones angulares tenga una solución de forma cerrada, es que tres ejes de articulación adyacentes se intersequen en un punto. En la sección 4.6 hablaremos sobre esta condición. Casi todos los manipuladores con seis grados de libertad construidos actualmente tienen tres ejes que se intersecan. Por ejemplo, los ejes 4, 5 y 6 del PUMA 560 se intersecan.

4.3 LA NOCIÓN DE SUBESPACIO DE MANIPULADOR CUANDO $N < 6$

El conjunto de tramas de destino alcanzables para un manipulador dado constituye su espacio de trabajo alcanzable. Para un manipulador con n grados de libertad (en donde $n < 6$), este espacio de trabajo alcanzable puede considerarse como una porción de un **subespacio** con n grados de libertad. De la misma forma en que el espacio de trabajo de un manipulador con seis grados de libertad es un subconjunto del espacio, el espacio de trabajo de un manipulador más simple es un subconjunto de su subespacio. Por ejemplo, el subespacio del robot de dos vínculos de la figura 4.1 es un plano, pero el espacio de trabajo es un subconjunto de este plano; es decir, un círculo de radio $l_1 + l_2$ para el caso en que $l_1 = l_2$.

Una manera de especificar el subespacio de un manipulador con n grados de libertad es proporcionar una expresión para su trama de muñeca o de herramienta, como función de n variables que la posicionan. Si consideramos que estas n variables son libres, entonces, a medida que vayan tomando todos los valores posibles, se generará el subespacio.

EJEMPLO 4.1

Proporcione una descripción del subespacio de ${}^B_WT$ para el manipulador de tres vínculos de la figura 3.6.

El subespacio de ${}^B_WT$ se da mediante

$$
{}^B_WT = \begin{bmatrix} c_\phi & -s_\phi & 0.0 & x \\ s_\phi & c_\phi & 0.0 & y \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.2}
$$

en donde x y y proporcionan la posición de la muñeca y ϕ describe la orientación del vínculo terminal. A medida que se permite a x, y y ϕ tomar valores arbitrarios, se genera el subespacio. Cualquier trama de muñeca que no tenga la estructura de (4.2) se encuentra fuera del subespacio (y por lo tanto queda fuera del espacio de trabajo) de este manipulador. Las longitudes de los vínculos y los límites de las articulaciones restringen el espacio de trabajo del manipulador para que sea un subconjunto de este subespacio.

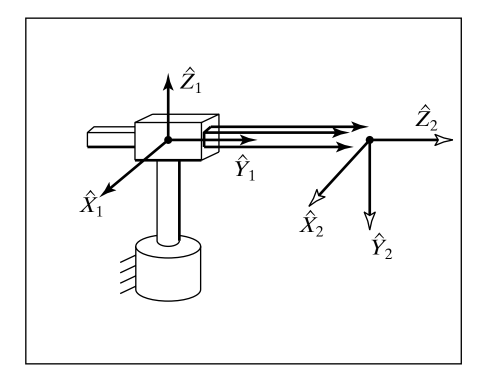


FIGURA 4.6: Un manipulador polar de dos vínculos.

EJEMPLO 4.2

Proporcione una descripción del subespacio de ${}^0_2T$ para el manipulador polar con dos grados de libertad que se muestra en la figura 4.6. Tenemos que

$$ {}^0P_{2ORG} = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix}, \tag{4.3} $$

en donde x y y pueden tomar cualquier valor. La orientación está restringida ya que el eje ${}^0\hat{Z}_2$ debe apuntar en una dirección que dependa de x y y. El eje ${}^0\hat{Y}_2$ debe apuntar siempre hacia abajo, y el eje ${}^0\hat{X}_2$ puede calcularse como el producto cruz de ${}^0\hat{Y}_2 \times {}^0\hat{Z}_2$. En términos de x y y tenemos que

$$ {}^0\hat{Z}_2 = \begin{bmatrix} \frac{x}{\sqrt{x^2+y^2}} \\ \frac{y}{\sqrt{x^2+y^2}} \\ 0 \end{bmatrix}. \tag{4.4} $$

El subespacio puede por lo tanto darse como

$$ {}^0_2T = \begin{bmatrix} \frac{y}{\sqrt{x^2+y^2}} & 0 & \frac{x}{\sqrt{x^2+y^2}} & x \\ \frac{-x}{\sqrt{x^2+y^2}} & 0 & \frac{y}{\sqrt{x^2+y^2}} & y \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.5} $$

Casi siempre, al definir un destino u objetivo para un manipulador con n grados de libertad, utilizamos n parámetros para especificar el destino. Si, por otra parte, proporcionamos una especificación de seis grados completos de libertad, no podremos en general alcanzar el destino con un manipulador $n < 6$. En este caso, podría ser conveniente alcanzar mejor un destino que se encuentre en el subespacio del manipulador y que esté lo más "cercano" posible al destino original deseado.

Por ende, al especificar destinos *generales* para un manipulador con menos de seis grados de libertad, una estrategia de solución es la siguiente:

1. Dada una trama de destino general ${}^S_GT$, calcule una trama de destino modificada ${}^S_{G'}T$ de tal manera que ${}^S_{G'}T$ se encuentre en el subespacio del manipulador y esté lo más "cerca posible" de ${}^S_GT$. Debe elegirse una definición de "cerca".
2. De tal manera la cinemática inversa para encontrar los ángulos de articulación utilizando ${}^S_{G'}T$ como el destino deseado. Considere que tal vez no sería posible tener aún una solución si el punto de destino no se encuentra en el espacio de trabajo del manipulador.

Casi siempre tiene sentido posicionar el origen de la trama herramienta en la ubicación deseada y luego seleccionar una orientación que pueda obtenerse y que esté cerca de la orientación deseada. Como vimos en los ejemplos 4.1 y 4.2, el cálculo del subespacio depende de la geometría del manipulador. Cada manipulador debe considerarse de manera individual para llegar a un método que realice este cálculo.

La sección 4.7 nos da un ejemplo de cómo *proyectar* un objetivo general en el subespacio de un manipulador con cinco grados de libertad para calcular los ángulos de articulación que hagan que el manipulador llegue a la trama más cercana a la deseada.

4.4 COMPARACIÓN DE LOS ENFOQUES ALGEBRAICO Y GEOMÉTRICO

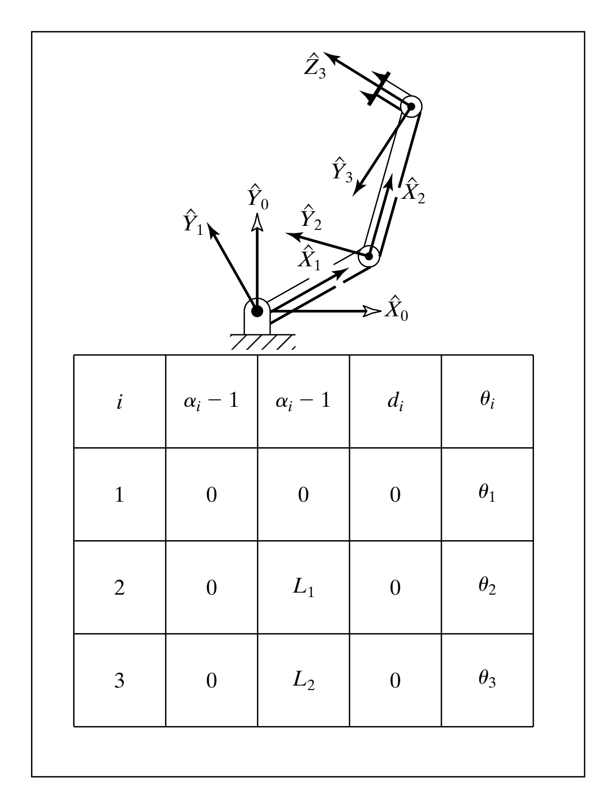
Como introducción a la solución de ecuaciones cinemáticas consideraremos dos métodos distintos para la solución de un manipulador planar simple de tres vínculos.

Solución algebraica

Considere el manipulador planar de tres vínculos que se presentó en el capítulo 3. Se muestra con sus parámetros de vínculo en la figura 4.7.

Si seguimos el método del capítulo 3 podemos usar los parámetros de vínculo fácilmente para encontrar las ecuaciones cinemáticas de este brazo:

$$ {}^{B}_{W}T = {}^{0}_{3}T = \begin{bmatrix} c_{123} & -s_{123} & 0.0 & l_1c_1 + l_2c_{12} \\ s_{123} & c_{123} & 0.0 & l_1s_1 + l_2s_{12} \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.6} $$



i	$\alpha_i - 1$	$\alpha_i - 1$	d_i	θ_i
1	0	0	0	θ_1
2	0	L_1	0	θ_2
3	0	L_2	0	θ_3

FIGURA 4.7: Manipulador planar de tres vínculos y sus parámetros correspondientes.

Para enfocar nuestra discusión en la cinemática inversa, vamos a suponer que se han realizado las transformaciones necesarias, de manera que el punto de destino sea una especificación de la trama de muñeca relativa a la trama base, la cual es ${}^{B}_{W}T$. Como estamos trabajando con un manipulador planar, puede lograrse la especificación de estos puntos de destino muy fácilmente, especificando tres números: x, y y ϕ, en donde ϕ es la orientación del vínculo 3 en el plano (relativo al eje $+\hat{X}$). Así, en vez de proporcionar una ${}^{B}_{W}T$ general como especificación de un destino, supondremos una transformación con la siguiente estructura:

$$
{}^{B}_{W}T = \begin{bmatrix} c_\phi & -s_\phi & 0.0 & x \\ s_\phi & c_\phi & 0.0 & y \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.7}
$$

Todos los destinos alcanzables deben encontrarse en el subespacio implicado por la estructura de la ecuación (4.7). Al igualar (4.6) y (4.7) llegamos a un conjunto de cuatro ecuaciones no lineales que deben resolverse para θ_1, θ_2 y θ_3:

$$c_\phi = c_{123}, \tag{4.8}$$

$$s_\phi = s_{123}, \tag{4.9}$$

$$x = l_1 c_1 + l_2 c_{12}, \tag{4.10}$$

$$y = l_1 s_1 + l_2 s_{12}. \tag{4.11}$$

Ahora comenzaremos nuestra solución algebraica de las ecuaciones (4.8) a la (4.11). Si elevamos al cuadrado las ecuaciones (4.10) y (4.11) y las sumamos, obtendremos:

$$x^2 + y^2 = l_1^2 + l_2^2 + 2 l_1 l_2 c_2, \tag{4.12}$$

en donde habremos hecho uso de

$$
\begin{aligned}
c_{12} &= c_1 c_2 - s_1 s_2, \\
s_{12} &= c_1 s_2 + s_1 c_2.
\end{aligned} \tag{4.13}
$$

Resolviendo (4.12) para c_2, obtendremos:

$$c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2}. \tag{4.14}$$

Para que pueda existir una solución, el lado derecho de (4.14) debe tener un valor entre -1 y 1. En el algoritmo de solución, esta restricción se comprobaría en este momento para averiguar si existe una solución. Físicamente, si esta restricción no se satisface, entonces el punto de destino está demasiado lejos como para que el manipulador lo alcance.

Suponiendo que el destino está en el espacio de trabajo, escribimos una expresión para s_2 de la siguiente manera:

$$s_2 = \pm\sqrt{1 - c_2^2}. \tag{4.15}$$

Por último, calculamos θ_2 con la rutina de arco tangente de dos argumentos:[1]

$$\theta_2 = \text{Atan2}(s_2, c_2). \tag{4.16}$$

[1] Vea la sección 2.8.

Esta opción de signos en la ecuación (4.15) corresponde a la solución múltiple en la que podemos elegir la solución de "codo hacia arriba" o de "codo hacia abajo". Para determinar θ_2 hemos usado uno de los métodos recurrentes para resolver el tipo de relaciones cinemáticas que surgen con más frecuencia: determinar tanto el seno como el coseno del ángulo de articulación deseado y luego aplicar la función arco tangente de dos argumentos. Esto asegura que encontremos todas las soluciones y que el ángulo resuelto se encuentre en el cuadrante apropiado.

Después de encontrar θ_2, podemos resolver las ecuaciones (4.10) y (4.11) para θ_1. Escribimos (4.10) y (4.11) de la siguiente forma:

$$x = k_1 c_1 - k_2 s_1, \tag{4.17}$$

$$y = k_1 s_1 + k_2 c_1, \tag{4.18}$$

en donde

$$k_1 = l_1 + l_2 c_2,$$
$$k_2 = l_2 s_2. \tag{4.19}$$

Para poder resolver una ecuación de esta forma realizamos un cambio de variables. En realidad estamos cambiando la forma en que escribimos las constantes k_1 y k_2.

Si

$$r = +\sqrt{k_1^2 + k_2^2} \tag{4.20}$$

y

$$\gamma = \text{Atan2}(k_2, k_1),$$

entonces

$$k_1 = r \cos \gamma,$$
$$k_2 = r \operatorname{sen} \gamma. \tag{4.21}$$

Las ecuaciones (4.17) y (4.18) pueden ahora escribirse así:

$$\frac{x}{r} = \cos \gamma \cos \theta_1 - \operatorname{sen} \gamma \operatorname{sen} \theta_1, \tag{4.22}$$

$$\frac{y}{r} = \cos \gamma \operatorname{sen} \theta_1 + \operatorname{sen} \gamma \cos \theta_1, \tag{4.23}$$

por lo tanto

$$\cos(\gamma + \theta_1) = \frac{x}{r}, \tag{4.24}$$

$$\operatorname{sen}(\gamma + \theta_1) = \frac{y}{r}. \tag{4.25}$$

Usando el arreglo de dos argumentos, obtenemos

$$\gamma + \theta_1 = \text{Atan2}\left(\frac{y}{r}, \frac{x}{r}\right) = \text{Atan2}(y, x), \tag{4.26}$$

y, por lo tanto

$$\theta_1 = \text{Atan2}(y, x) - \text{Atan2}(k_2, k_1). \tag{4.27}$$

Observe que cuando se elige un signo en la solución de θ_2 descrita anteriormente, se provoca un cambio de signo en k_2, lo cual afecta a θ_1. Las sustituciones (4.20) y (4.21) utilizadas constituyen un método de solución de una forma que aparece frecuentemente en cinemática: la de las ecuaciones (4.10) o (4.11). Observe también que si $x = y = 0$, entonces la ecuación (4.27) se vuelve indefinida y, en este caso, el valor de θ_1 es arbitrario.

Finalmente, de las ecuaciones (4.8) y (4.9) podemos resolver para la suma de θ_1 a θ_3:

$$\theta_1 + \theta_2 + \theta_3 = \text{Atan2}(s_\phi, c_\phi) = \phi. \tag{4.28}$$

De este resultado, podemos despejar θ_3, ya que conocemos los primeros dos ángulos. Es común con los manipuladores que tienen dos o más vínculos que se mueven en un plano que, en el curso de la solución, surjan expresiones para las sumas de los ángulos de articulación.

En resumen, un método algebraico para resolver ecuaciones cinemáticas es básicamente el de manipular las ecuaciones dadas en una forma para la cual se conozca una solución. Resulta que para muchas geometrías comunes con frecuencia surgen varias formas de ecuaciones trascendentales. Hemos encontrado un par de ellas en la sección anterior. (En el apéndice C se muestran más de estas ecuaciones).

Solución geométrica

En un método geométrico para encontrar la solución de un manipulador, tratamos de descomponer la geometría espacial del brazo en varios problemas de geometría plana. Para muchos manipuladores (especialmente cuando las $\alpha_i = 0$ o ± 90), esto puede hacerse con mucha facilidad. Los ángulos de articulación pueden así resolverse utilizando las herramientas de la geometría plana [7]. Para el brazo con tres grados de libertad que se muestra en la figura 4.7, como dicho brazo es planar podemos aplicar directamente la geometría plana para encontrar una solución.

La figura 4.8 muestra el triángulo formado por l_1, l_2 y la línea que une el origen de la trama {0} con el origen de la trama {3}. Las líneas punteadas representan otra posible configuración del triángulo, la cual llevaría a la misma posición de la trama {3}. Considerando el triángulo sólido, podemos aplicar la "ley de los cosenos" y resolver para θ_2:

$$x^2 + y^2 = l_1^2 + l_2^2 - 2 l_1 l_2 \cos(180 + \theta_2). \tag{4.29}$$

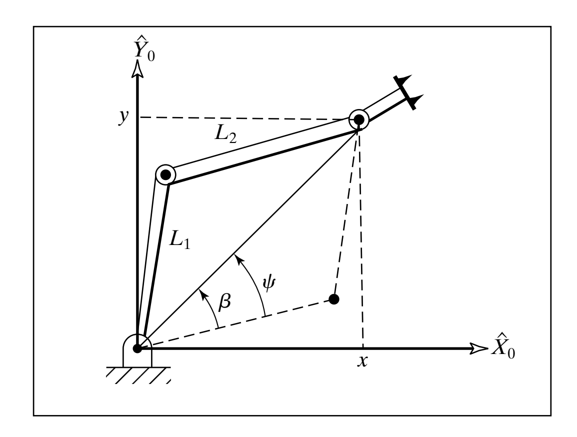


FIGURA 4.8: Geometría plana asociada con un robot planar de tres vínculos.

Ahora; $\cos(180 + \theta_2) = -\cos(\theta_2)$, por lo que tenemos:

$$c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2}. \tag{4.30}$$

Para que este triángulo exista, la distancia al punto de destino $\sqrt{x^2 + y^2}$ debe ser menor o igual que la suma de las longitudes de los vínculos, $l_1 + l_2$. Esta condición se comprobaría en este punto en un algoritmo computacional para verificar la existencia de soluciones. Esta condición no se satisface cuando el punto de destino está fuera del alcance del manipulador. Suponiendo que exista una solución, esta ecuación se resuelve para aquel valor de θ_2 que se encuentre entre 0 y -180 grados, ya que sólo para estos valores existe el triángulo en la figura 4.8. La otra posible solución (la que está indicada por el triángulo de líneas punteadas) se encuentra por simetría como $\theta'_2 = -\theta_2$.

Para despejar θ_1 encontramos expresiones para los ángulos ψ y β según lo indicado en la figura 4.8. En primer lugar β puede estar en cualquier cuadrante, dependiendo de los signos de x y de y. Por ello debemos usar una función arco tangente de dos argumentos:

$$\beta = \text{Atan2}(y, x). \tag{4.31}$$

De nuevo aplicamos la ley de los cosenos para encontrar ψ:

$$\cos\psi = \frac{x^2 + y^2 + l_1^2 - l_2^2}{2l_1\sqrt{x^2 + y^2}}. \tag{4.32}$$

Aquí el arco coseno debe resolverse de manera que $0 \leq \psi \leq 180°$, para poder preservar la geometría que nos lleve a la ecuación (4.32). Estas consideraciones son típicas cuando se utiliza un método geométrico; debemos aplicar las fórmulas que derivamos solamente sobre un rango de variables tal que se preserve la geometría. Entonces tenemos que:

$$\theta_1 = \beta \pm \psi, \tag{4.33}$$

en donde se utiliza el signo positivo si $\theta_2 < 0$ y el signo negativo si $\theta_2 > 0$.

Sabemos que los ángulos en un plano se suman, por lo que la suma de los tres ángulos de articulación debe ser la orientación del último vínculo:

$$\theta_1 + \theta_2 + \theta_3 = \phi. \tag{4.34}$$

Esta ecuación se resuelve para θ_3, para completar nuestra solución.

4.5 SOLUCIÓN ALGEBRAICA POR REDUCCIÓN A POLINOMIO

Las ecuaciones trascendentales son a menudo difíciles de resolver porque, aun y cuando solamente hay una variable (digamos, θ), generalmente aparece como sen θ y cos θ. No obstante, si hacemos las siguientes sustituciones se produce una expresión en términos de una sola variable u:

$$u = \tan\frac{\theta}{2},$$
$$\cos\theta = \frac{1 - u^2}{1 + u^2}, \tag{4.35}$$
$$\text{sen}\,\theta = \frac{2u}{1 + u^2}.$$

Ésta es una sustitución geométrica muy importante que se utiliza a menudo para resolver ecuaciones cinemáticas. Estas sustituciones convierten ecuaciones trascendentales en ecuaciones polinomiales en u. El apéndice A contiene una lista de éstas y otras identidades trigonométricas.

EJEMPLO 4.3

Convierta la ecuación trascendental

$$a\cos\theta + b\,\mathrm{sen}\,\theta = c \tag{4.36}$$

en una ecuación polinomial en la tangente del medio ángulo, y resuelva para θ.

Sustituyendo de la ecuación (4.35) y multiplicando por $1 + u^2$, tenemos que

$$a(1 - u^2) + 2bu = c(1 + u^2). \tag{4.37}$$

Si juntamos las potencias de u tenemos que

$$(a + c)u^2 - 2bu + (c - a) = 0, \tag{4.38}$$

que se resuelve mediante la fórmula cuadrática:

$$u = \frac{b \pm \sqrt{b^2 + a^2 - c^2}}{a + c}. \tag{4.39}$$

Por lo tanto,

$$\theta = 2\tan^{-1}\left(\frac{b \pm \sqrt{b^2 + a^2 - c^2}}{a + c}\right). \tag{4.40}$$

En caso de que la solución para u en base a la ecuación (4.39) fuera compleja, no hay solución real para la ecuación trascendental original. Observe que si $a + c = 0$, el argumento del arco tangente se vuelve infinito y por ende $\theta = 180°$. En una implementación por computadora debe comprobarse antes esta división potencial entre cero. Esta situación se produce cuando el término cuadrático de (4.38) desaparece, de manera que la ecuación cuadrática se degenera en una ecuación lineal.

Los polinomios de hasta cuarto grado poseen soluciones de forma cerrada [8, 9], por lo que los manipuladores que sean lo suficientemente simples como para poder resolverse mediante ecuaciones algebraicas de este grado (o menores) se llaman manipuladores **con solución de forma cerrada**.

4.6 SOLUCIÓN DE PIEPER CUANDO SE INTERSECAN TRES EJES

Como se mencionó anteriormente, aunque un robot completamente general con seis grados de libertad no tenga una solución de forma cerrada, pueden resolverse ciertos casos especiales importantes. Pieper [3, 4] estudió los manipuladores con seis grados de libertad en los cuales tres ejes consecutivos se intersecan en un punto.[2] En esta sección describiremos el método que él desarrolló para el caso de las seis articulaciones angulares, en donde los últimos tres ejes se intersecan. Su método se aplica a otras configuraciones que incluyen articulaciones prismáticas, por lo que el lector interesado debería

---

[2] En esta familia de manipuladores se incluyen los que tienen tres ejes paralelos consecutivos, ya que se encuentran en el punto que está en el infinito.

leer la referencia [4]. El trabajo de Pieper se aplica a la mayoría de los robots industriales comercialmente disponibles.

Cuando se intersecan los últimos tres ejes, los orígenes de las tramas de vínculo {4}, {5} y {6} se ubican en este punto de intersección, el cual se da en coordenadas base de la siguiente manera:

$$ {}^{0}P_{4ORG} = {}^{0}_{1}T \, {}^{1}_{2}T \, {}^{2}_{3}T \, {}^{3}P_{4ORG} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \tag{4.41} $$

o, utilizando la cuarta columna de la ecuación (3.6) para $i = 4$, como:

$$ {}^{0}P_{4ORG} = {}^{0}_{1}T \, {}^{1}_{2}T \, {}^{2}_{3}T \begin{bmatrix} a_3 \\ -d_4 s\alpha_3 \\ d_4 c\alpha_3 \\ 1 \end{bmatrix}, \tag{4.42} $$

o como

$$ {}^{0}P_{4ORG} = {}^{0}_{1}T \, {}^{1}_{2}T \begin{bmatrix} f_1(\theta_3) \\ f_2(\theta_3) \\ f_3(\theta_3) \\ 1 \end{bmatrix}, \tag{4.43} $$

en donde

$$ \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ 1 \end{bmatrix} = {}^{2}_{3}T \begin{bmatrix} a_3 \\ -d_4 s\alpha_3 \\ d_4 c\alpha_3 \\ 1 \end{bmatrix}. \tag{4.44} $$

Utilizando la ecuación (3.6) para ${}^{2}_{3}T$ en la ecuación (4.44) se producen las siguientes expresiones para f_i:

$$ \begin{aligned} f_1 &= a_3 c_3 + d_4 s\alpha_3 s_3 + a_2, \\ f_2 &= a_3 c\alpha_2 s_3 - d_4 s\alpha_3 c\alpha_2 c_3 - d_4 s\alpha_2 c\alpha_3 - d_3 s\alpha_2, \\ f_3 &= a_3 s\alpha_2 s_3 - d_4 s\alpha_3 s\alpha_2 c_3 + d_4 c\alpha_2 c\alpha_3 + d_3 c\alpha_2. \end{aligned} \tag{4.45} $$

Utilizando la ecuación (3.6) para ${}^{0}_{1}T$ y ${}^{1}_{2}T$ en la ecuación (4.43), obtenemos

$$ {}^{0}P_{4ORG} = \begin{bmatrix} c_1 g_1 - s_1 g_2 \\ s_1 g_1 + c_1 g_2 \\ g_3 \\ 1 \end{bmatrix}, \tag{4.46} $$

en donde

$$ \begin{aligned} g_1 &= c_2 f_1 - s_2 f_2 + a_1, \\ g_2 &= s_2 c\alpha_1 f_1 + c_2 c\alpha_1 f_2 - s\alpha_1 f_3 - d_2 s\alpha_1, \\ g_3 &= s_2 s\alpha_1 f_1 + c_2 s\alpha_1 f_2 + c\alpha_1 f_3 + d_2 c\alpha_1. \end{aligned} \tag{4.47} $$

Ahora escriba una expresión para la magnitud al cuadrado de ${}^{0}P_{4ORG}$, que denotaremos como $r = x^2 + y^2 + z^2$, y que por la ecuación (4.46) puede verse como

$$ r = g_1^2 + g_2^2 + g_3^2; \tag{4.48} $$

por lo que, utilizando la ecuación (4.47) para las g_i, tenemos que

$$r = f_1^2 + f_2^2 + f_3^2 + a_1^2 + d_2^2 + 2d_2 f_3 + 2a_1(c_2 f_1 - s_2 f_2). \tag{4.49}$$

Ahora escribiremos esta ecuación, junto con la del componente Z de (4.46), como un sistema de dos ecuaciones en la forma

$$\begin{aligned} r &= (k_1 c_2 + k_2 s_2)2a_1 + k_3, \\ z &= (k_1 s_2 - k_2 c_2)s\alpha_1 + k_4, \end{aligned} \tag{4.50}$$

en donde

$$\begin{aligned} k_1 &= f_1, \\ k_2 &= -f_2, \\ k_3 &= f_1^2 + f_2^2 + f_3^2 + a_1^2 + d_2^2 + 2d_2 f_3, \\ k_4 &= f_3 c\alpha_1 + d_2 c\alpha_1. \end{aligned} \tag{4.51}$$

La ecuación (4.50) es útil, ya que se ha eliminado la dependencia de θ_1 y a que la dependencia de θ_2 toma una forma simple.

Ahora consideremos la solución de la ecuación (4.50) para θ_3. Distinguimos tres casos:

1. Si $a_1 = 0$, tenemos que $r = k_3$, en donde se conoce r. El lado derecho (k_3) es una función de θ_3 solamente. Después de la sustitución en (4.35), puede resolverse una ecuación cuadrática en tan $\frac{\theta_3}{2}$ para θ_3.
2. Si $s\alpha_1 = 0$, tenemos que $z = k_4$, en donde se conoce z. De nuevo, después de sustituir mediante la ecuación (4.35), surge una ecuación cuadrática que puede resolverse para θ_3.
3. De cualquier otra forma, elimine s_2 y c_2 de la ecuación (4.50) para obtener

$$\frac{(r - k_3)^2}{4a_1^2} + \frac{(z - k_4)^2}{s^2\alpha_1} = k_1^2 + k_2^2. \tag{4.52}$$

Esta ecuación, después de sustituir (4.35) para θ_3, produce una ecuación de grado 4, la cual puede resolverse para θ_3.[3]

Habiendo resuelto para θ_3, podemos resolver la ecuación (4.50) para θ_2 y la (4.46) para θ_1.

Para completar nuestra solución, necesitamos resolver para θ_4, θ_5 y θ_6. Estos ejes se intersecan, por lo que estos ángulos de articulación afectan a la orientación del último vínculo solamente. Podemos calcularlos a partir únicamente de la porción de rotación del destino especificado, ${}^0_6R$. Después de obtener θ_1, θ_2 y θ_3, podemos calcular ${}^0_4R|_{\theta 4=0}$, por cuya notación indicamos la orientación de la trama de vínculo {4} relativa a la trama base cuando $\theta_4 = 0$. La orientación deseada de {6} difiere de esta orientación sólo por la acción de las últimas tres articulaciones. Como el problema se especificó bajo el supuesto de que se conoce ${}^0_6R$, podemos calcular

$${}^4_6R|_{\theta_4=0} = {}^0_4R^{-1}|_{\theta_4=0}\, {}^0_6R. \tag{4.53}$$

[3] Es útil tener en cuenta que $f_1^2 + f_2^2 + f_3^2 = a_3^2 + d_4^2 + d_3^2 + a_2^2 + 2d_4 d_3 c\alpha_3 + 2a_2 a_3 c_3 + 2a_2 d_4 s\alpha_3 s_3$.

Para muchos manipuladores, estos tres últimos ángulos pueden despejarse utilizando exactamente la solución de ángulos de Euler Z-Y-Z dada en el capítulo 2, aplicada a ${}^0_4R|_{\theta 4=0}$. Para cualquier manipulador (en el que se intersequen los ejes 4, 5 y 6), los últimos tres ángulos de articulación pueden resolverse como un conjunto de ángulos de Euler definidos apropiadamente. Siempre hay dos soluciones para estas últimas tres articulaciones, por lo que el número total de soluciones para el manipulador será el doble del número encontrado para las primeras tres articulaciones.

4.7 EJEMPLOS DE CINEMÁTICA INVERSA DE MANIPULADORES

En esta sección trabajaremos con la cinemática inversa de dos robots industriales. Una solución para manipuladores se realizará de manera puramente algebraica; la segunda solución es en parte algebraica y en parte geométrica. Las siguientes soluciones no constituyen un recetario para resolver la cinemática de manipuladores, pero muestran muchas de las manipulaciones comunes que comúnmente aparecen en la mayoría de las soluciones de cinemática. Tenga en cuenta que puede usar el método Pieper de solución (cubierto en la sección anterior) para estos manipuladores, pero aquí optamos por llegar a la solución de una manera distinta para entender mejor los diversos métodos disponibles.

El robot Unimation PUMA 560

Como ejemplo de la técnica de solución algebraica aplicada a un manipulador con seis grados de libertad, resolveremos las ecuaciones cinemáticas del PUMA 560 que se desarrollaron en el capítulo 3. Esta solución está en el estilo de [5].

Deseamos resolver

$$\begin{aligned} {}^0_6T &= \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= {}^0_1T(\theta_1)\,{}^1_2T(\theta_2)\,{}^2_3T(\theta_3)\,{}^3_4T(\theta_4)\,{}^4_5T(\theta_5)\,{}^5_6T(\theta_6) \end{aligned} \tag{4.54}$$

para θ_i cuando se da ${}^0_6T$ en forma de valores numéricos.

Un nuevo planteamiento de la ecuación (4.54) que pone la dependencia en θ_1 del lado izquierdo de la ecuación es

$$[{}^0_1T(\theta_1)]^{-1}\,{}^0_6T = {}^1_2T(\theta_2)\,{}^2_3T(\theta_3)\,{}^3_4T(\theta_4)\,{}^4_5T(\theta_5)\,{}^5_6T(\theta_6). \tag{4.55}$$

Si invertimos ${}^0_1T$, escribimos la ecuación (4.55) como

$$\begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^1_6T, \tag{4.56}$$

en donde ${}^1_6T$ se da mediante la ecuación (3.13) que se desarrolló en el capítulo 3. Esta técnica simple de multiplicar cada lado de una ecuación de transformadas por una inversa se utiliza a menudo para aprovechar la separación de las variables en la búsqueda de una ecuación que pueda resolverse.

Si igualamos los elementos (2,4) de ambos lados de la ecuación (4.56), tendremos que

$$-s_1 p_x + c_1 p_y = d_3. \tag{4.57}$$

Para resolver una ecuación de esta forma, hacemos las siguientes sustituciones trigonométricas:

$$\begin{aligned} p_x &= \rho \cos\phi, \\ p_y &= \rho \operatorname{sen}\phi, \end{aligned} \tag{4.58}$$

en donde

$$\begin{aligned} \rho &= \sqrt{p_x^2 + p_y^2}, \\ \phi &= \operatorname{Atan2}(p_y, p_x). \end{aligned} \tag{4.59}$$

Si sustituimos la ecuación (4.58) en la (4.57), obtenemos

$$c_1 s_\phi - s_1 c_\phi = \frac{d_3}{\rho}. \tag{4.60}$$

De la fórmula de diferencia de ángulos,

$$\operatorname{sen}(\phi - \theta_1) = \frac{d_3}{\rho}. \tag{4.61}$$

Por lo tanto,

$$\cos(\phi - \theta_1) = \pm\sqrt{1 - \frac{d_3^2}{\rho^2}}, \tag{4.62}$$

y también

$$\phi - \theta_1 = \operatorname{Atan2}\left(\frac{d_3}{\rho}, \pm\sqrt{1 - \frac{d_3^2}{\rho^2}}\right). \tag{4.63}$$

Por último, la solución para θ_1 puede escribirse como

$$\theta_1 = \operatorname{Atan2}(p_y, p_x) - \operatorname{Atan2}\left(d_3, \pm\sqrt{p_x^2 + p_y^2 - d_3^2}\right). \tag{4.64}$$

Observe que hemos encontrado dos posibles soluciones para θ_1, que corresponden al signo positivo o negativo en la ecuación (4.64). Ahora que tenemos θ_1, se conoce el lado izquierdo de la ecuación (4.56). Si igualamos tanto los elementos (1,4) como (3,4) de los dos lados de (4.56), obtendremos

$$\begin{aligned} c_1 p_x + s_1 p_y &= a_3 c_{23} - d_4 s_{23} + a_2 c_2, \\ -p_x &= a_3 s_{23} + d_4 c_{23} + a_2 s_2. \end{aligned} \tag{4.65}$$

Si elevamos al cuadrado las ecuaciones (4.65) y (4.57) y sumamos las ecuaciones resultantes, obtendremos

$$a_3 c_3 - d_4 s_3 = K, \tag{4.66}$$

en donde

$$K = \frac{p_x^2 + p_y^2 + p_x^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2}. \tag{4.67}$$

Observe que se ha quitado la dependencia de θ_1 de la ecuación (4.66). Esta ecuación tiene la misma forma que la ecuación (4.57), por lo que puede resolverse por el mismo tipo de sustitución trigonométrica para generar una solución para θ_3:

$$\theta_3 = \text{Atan2}(a_3, d_4) - \text{Atan2}(K, \pm\sqrt{a_3^2 + d_4^2 - K^2}). \tag{4.68}$$

El signo positivo o negativo en la ecuación (4.68) conduce a dos distintas soluciones para θ_3. Si consideramos de nuevo la ecuación (4.54), ahora podemos reescribirla de manera que todo el lado izquierdo sea una función solamente de variables conocidas y de θ_2:

$$[{}^0_3T(\theta_2)]^{-1}\,{}^0_6T = {}^3_4T(\theta_4)\,{}^4_5T(\theta_5)\,{}^5_6T(\theta_6), \tag{4.69}$$

o

$$\begin{bmatrix} c_1c_{23} & s_1c_{23} & -s_{23} & -a_2c_3 \\ -c_1s_{23} & -s_1s_{23} & -c_{23} & a_2s_3 \\ -s_1 & c_1 & 0 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^3_6T, \tag{4.70}$$

en donde ${}^3_6T$ se da mediante la ecuación (3.11) desarrollada en el capítulo 3. Igualando los elementos (1,4) y (2,4) de los dos lados de la ecuación (4.70), obtenemos

$$\begin{aligned} c_1c_{23}p_x + s_1c_{23}p_y - s_{23}p_z - a_2c_3 &= a_3, \\ -c_1s_{23}p_x - s_1s_{23}p_y - c_{23}p_z + a_2s_3 &= d_4. \end{aligned} \tag{4.71}$$

Estas ecuaciones pueden resolverse simultáneamente para s_{23} y c_{23}, lo que produce

$$\begin{aligned} s_{23} &= \frac{(-a_3 - a_2c_3)p_z + (c_1p_x + s_1p_y)(a_2s_3 - d_4)}{p_z^2 + (c_1p_x + s_1p_y)^2}, \\ c_{23} &= \frac{(a_2s_3 - d_4)p_z - (a_3 + a_2c_3)(c_1p_x + s_1p_y)}{p_z^2 + (c_1p_x + s_1p_y)^2}. \end{aligned} \tag{4.72}$$

Los denominadores son iguales y positivos, por lo que resolvemos para la suma de θ_2 y θ_3 como

$$\begin{aligned} \theta_{23} = \text{Atan2}[&(-a_3 - a_2c_3)p_z - (c_1p_x + s_1p_y)(d_4 - a_2s_3), \\ &(a_2s_3 - d_4)p_z - (a_3 + a_2c_3)(c_1p_x + s_1p_y)]. \end{aligned} \tag{4.73}$$

La ecuación (4.73) calcula cuatro valores de θ_{23}, de acuerdo con las cuatro posibles combinaciones de las soluciones para θ_1 y θ_3; por lo que las cuatro posibles soluciones para θ_2 se pueden calcular así:

$$\theta_2 = \theta_{23} - \theta_3, \tag{4.74}$$

en donde la solución apropiada para θ_3 se obtiene al formar la diferencia.

Ahora conocemos todo el lado izquierdo de la ecuación (4.70). Igualando los elementos (1,3) y (3,3) de los dos lados de (4.70) obtenemos

$$\begin{aligned} r_{13}c_1c_{23} + r_{23}s_1c_{23} - r_{33}s_{23} &= -c_4s_5, \\ -r_{13}s_1 + r_{23}c_1 &= s_4s_5. \end{aligned} \tag{4.75}$$

Siempre y cuando $s_5 \neq 0$, podremos resolver para θ_4 de la siguiente manera:

$$\theta_4 = \text{Atan2}(-r_{13}s_1 + r_{23}c_1, -r_{13}c_1c_{23} - r_{23}s_1c_{23} + r_{33}s_{23}). \tag{4.76}$$

Cuando $\theta_5 = 0$, el manipulador se encuentra en una configuración singular en la que los ejes de articulación 4 y 6 se alinean y producen el mismo movimiento del último vínculo del robot. En este caso, todo lo que importa (y todo lo que puede resolverse) es la suma o diferencia de θ_4 y θ_6. Esta situación se detecta comprobando si ambos argumentos de la función Atan2 en la ecuación (4.76) están cerca de cero. De ser así, θ_4 se elige arbitrariamente[4] y, cuando se calcule θ_6 posteriormente, se calculará acorde con ello.

Si vemos de nuevo la ecuación (4.56), podremos reescribirla ahora de manera que todo el lado izquierdo sea una función solamente de variables conocidas y θ_4; y la reescribiríamos, como

$$[{}^0_4T(\theta_4)]^{-1}\,{}^0_6T = {}^4_5T(\theta_5)\,{}^5_6T(\theta_6), \tag{4.77}$$

en donde $[{}^0_4T(\theta 4)]^{-1}$ se da mediante

$$\begin{bmatrix} c_1c_{23}c_4 + s_1s_4 & s_1c_{23}c_4 - c_1s_4 & -s_{23}c_4 & -a_2c_3c_4 + d_3s_4 - a_3c_4 \\ -c_1c_{23}s_4 + s_1c_4 & -s_1c_{23}s_4 - c_1c_4 & s_{23}s_4 & a_2c_3s_4 + d_3c_4 + a_3s_4 \\ -c_1s_{23} & -s_1s_{23} & -c_{23} & a_2s_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.78}$$

y ${}^4_6T$ mediante la ecuación (3.10) desarrollada en el capítulo 3. Igualando los elementos (1,3) y (3,3) de ambos lados de la ecuación (4.77), obtenemos

$$\begin{aligned} r_{13}(c_1c_{23}c_4 + s_1s_4) + r_{23}(s_1c_{23}c_4 - c_1s_4) - r_{33}(s_{23}c_4) &= -s_5, \\ r_{13}(-c_1s_{23}) + r_{23}(-s_1s_{23}) + r_{33}(-c_{23}) &= c_5. \end{aligned} \tag{4.79}$$

Por lo tanto, podemos resolver para θ_5 de la siguiente manera:

$$\theta_5 = \text{Atan2}(s_5, c_5), \tag{4.80}$$

en donde s_5 y c_5 se obtienen mediante la ecuación (4.79).

Aplicando el mismo método una vez más, calculamos $({}^0_5T)^{-1}$ y escribimos la ecuación (4.54) de la siguiente forma:

$$({}^0_5T)^{-1}\,{}^0_6T = {}^5_6T(\theta_6). \tag{4.81}$$

Igualando los elementos (3,1) y (1,1) de ambos lados de la ecuación (4.77), como lo hicimos antes, obtenemos

$$\theta_6 = \text{Atan2}(s_6, c_6), \tag{4.82}$$

en donde

$$\begin{aligned} s_6 &= -r_{11}(c_1c_{23}s_4 - s_1c_4) - r_{21}(s_1c_{23}s_4 + c_1c_4) + r_{31}(s_{23}s_4), \\ c_6 &= r_{11}[(c_1c_{23}c_4 + s_1s_4)c_5 - c_1s_{23}s_5] + r_{21}[(s_1c_{23}c_4 - c_1s_4)c_5 - s_1s_{23} \\ &\quad -r_{31}(s_{23}c_4c_5 + c_{23}s_5). \end{aligned}$$

Debido a los signos positivo y negativo que aparecen en las ecuaciones (4.64) y (4.68), éstas cuatro soluciones. Además, hay cuatro soluciones adicionales que se obtienen

---

[4] Por lo general se elige de manera que sea igual al valor actual de la articulación 4.

"volteando" la muñeca del manipulador. Para cada una de las cuatro soluciones calculadas antes, obtenemos la solución inversa mediante

$$\begin{aligned} \theta'_4 &= \theta_4 + 180°, \\ \theta'_5 &= -\theta_5, \\ \theta'_6 &= \theta_6 + 180°. \end{aligned} \tag{4.83}$$

Una vez que se han calculado las ocho soluciones, habrá que descartar algunas de ellas (o incluso todas) debido a violaciones en los límites de las articulaciones. De cualquier solución válida restante, generalmente se selecciona la más cercana a la configuración actual del manipulador.

El Yasukawa Motoman L-3

Como segundo ejemplo resolveremos las ecuaciones cinemáticas del Yasukawa Motoman L-3 que se desarrollaron en el capítulo 3. Esta solución será en parte algebraica y en parte geométrica. El Motoman L-3 tiene tres características que hacen que el problema de cinemática inversa sea bastante distinto al de PUMA. En primer lugar el manipulador sólo tiene cinco articulaciones, por lo que no puede posicionar y orientar su efector final para llegar a tramas que sean destinos *generales*. En segundo lugar, el tipo de articulaciones de cuatro barras y el esquema de transmisión de cadena hacen que un actuador mueva dos o más articulaciones. En tercer lugar, los límites de posición de cada actuador no son constantes, sino que dependen de las posiciones de los otros actuadores, por lo que no es fácil averiguar si un conjunto calculado de valores del actuador está en el rango.

Si consideramos la naturaleza del subespacio del manipulador Motoman (lo mismo se aplica a muchos manipuladores con cinco grados de libertad), rápidamente nos daremos cuenta que este subespacio puede describirse proporcionando una restricción en la orientación alcanzable: la dirección en la que apunta la herramienta, es decir, el eje $\hat{Z}_T$, debe estar en el "plano del brazo". Este plano es el plano vertical que contiene el eje de la articulación 1 y el punto en el que los ejes 4 y 5 se intersecan. La orientación más cercana a una orientación general es la que se obtiene girando la dirección en la que apunta la herramienta de manera que se encuentre en el plano, utilizando una mínima cantidad de rotación. Sin desarrollar una expresión explícita para este subespacio construiremos un método para proyectar una trama de destino general sobre él. Observe que toda esta discusión es para el caso en el que la trama de muñeca y la trama herramienta difieran solamente por una traslación sobre el eje $\hat{Z}_W$.

En la figura 4.9, indicamos el plano del brazo mediante su normal $\hat{M}$, y la dirección en la que se desea que apunte la herramienta mediante $\hat{Z}_T$. Esta dirección debe girarse por un ángulo θ sobre un vector $\hat{K}$ para poder hacer que la nueva dirección a la que apunte la herramienta, $\hat{Z}'_T$, se encuentre en el plano. Está claro que el vector $\hat{K}$ que minimiza θ se encuentra en el plano y es ortogonal a $\hat{Z}_T$ y $\hat{Z}'_T$.

Para cualquier trama de destino dada, $\hat{M}$ se define como

$$\hat{M} = \frac{1}{\sqrt{p_x^2 + p_y^2}} \begin{bmatrix} -p_y \\ p_x \\ 0 \end{bmatrix}, \tag{4.84}$$

en donde p_x y p_y son las coordenadas X y Y de la posición deseada para la herramienta. Entonces $\hat{K}$ se da mediante

$$\hat{K} = \hat{M} \times \hat{Z}_T. \tag{4.85}$$

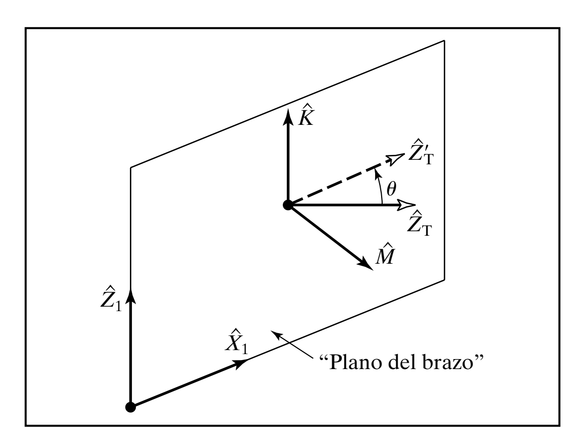


FIGURA 4.9: Cómo girar una trama de destino en el subespacio del Motoman.

El nuevo $\hat{Z}'_T$ sería

$$\hat{Z}'_T = \hat{K} \times \hat{M}. \tag{4.86}$$

La cantidad de rotación θ se da mediante

$$\cos\theta = \hat{Z}_T \cdot \hat{Z}'_T,$$
$$\operatorname{sen}\theta = (\hat{Z}_T \times \hat{Z}'_T) \cdot \hat{K}. \tag{4.87}$$

Usando la fórmula de rotación de Rodrigues (vea el ejercicio 2.20) tenemos que

$$\hat{Y}'_T = c\theta\hat{Y}_T + s\theta(\hat{K} \times \hat{Y}_T) + (1 - c\theta)(\hat{K} \cdot \hat{Y}_T)\hat{K}. \tag{4.88}$$

Por último, calculamos la columna desconocida restante de la nueva matriz de rotación de la herramienta de la siguiente manera:

$$\hat{X}'_T = \hat{Y}'_T \times \hat{Z}'_T. \tag{4.89}$$

Las ecuaciones (4.84) a (4.89) describen un método para proyectar una orientación de destino general dada en el subespacio del robot Motoman.

Suponiendo que la trama de muñeca ${}^B_W T$ dada se encuentre en el subespacio del manipulador, resolveremos las ecuaciones cinemáticas como se muestra a continuación. Al derivar las ecuaciones cinemáticas para el Motoman L-3 formamos el producto de transformaciones de vínculos:

$${}^0_5T = {}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_4T\,{}^4_5T. \tag{4.90}$$

Si dejamos que

$${}^0_5T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.91}$$

y multiplicamos previamente ambos lados por ${}^0_1T^{-1}$, tenemos

$${}^0_1T^{-1}\,{}^0_5T = {}^1_2T\,{}^2_3T\,{}^3_4T\,{}^4_5T, \tag{4.92}$$

en donde el lado izquierdo sería

$$\begin{bmatrix} c_1r_{11}+s_1r_{21} & c_1r_{12}+s_1r_{22} & c_1r_{13}+s_1r_{23} & c_1p_x+s_1p_y \\ -r_{31} & -r_{32} & -r_{33} & -p_z \\ -s_1r_{11}+c_1r_{21} & -s_1r_{12}+c_1r_{22} & -s_1r_{13}+c_1r_{23} & -s_1p_x+c_1p_y \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.93}$$

y el lado derecho sería

$$\begin{bmatrix} * & * & s_{234} & * \\ * & * & -c_{234} & * \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \tag{4.94}$$

en la ecuación anterior no se mostraron varios elementos. Si igualamos los elementos (3,4) obtendremos

$$-s_1p_x+c_1p_y=0, \tag{4.95}$$

lo cual nos produce[5]

$$\theta_1=\text{Atan2}(p_y,p_x). \tag{4.96}$$

Igualando los elementos (3,1) y (3,2) obtendremos

$$\begin{aligned} s_5 &= -s_1r_{11}+c_1r_{21}, \\ c_5 &= -s_1r_{12}+c_1r_{22}, \end{aligned} \tag{4.97}$$

de lo cual podemos calcular θ_5 de la siguiente manera:

$$\theta_5=\text{Atan2}(r_{21}c_1-r_{11}s_1,r_{22}c_1-r_{12}s_1). \tag{4.98}$$

Igualando los elementos (2,3) y (1,3) obtendremos

$$\begin{aligned} c_{234} &= r_{33}, \\ s_{234} &= c_1r_{13}+s_1r_{23}, \end{aligned} \tag{4.99}$$

lo cual nos conduce a

$$\theta_{234}=\text{Atan2}(r_{13}c_1+r_{23}s_1,r_{33}). \tag{4.100}$$

Para resolver los ángulos individuales θ_2, θ_3, y θ_4, usaremos un método geométrico. La figura 4.10 muestra el plano del brazo con el punto A en el eje de articulación 2, el punto B en el eje de articulación 3 y el punto C en el eje de articulación 4.

De la ley de los cosenos aplicada al triángulo ABC, tenemos que

$$\cos\theta_3=\frac{p_x^2+p_y^2+p_z^2-l_2^2-l_3^2}{2l_2l_3}. \tag{4.101}$$

Después tenemos que[6]

$$\theta_3=\text{Atan2}\left(\sqrt{1-\cos^2\theta_3},\cos\theta_3\right). \tag{4.102}$$

[5] Para este manipulador, una segunda solución violaría los límites de las articulaciones, por lo que no se calcula.

[6] Para este manipulador, una segunda solución violaría los límites de las articulaciones, por lo que no se calcula.

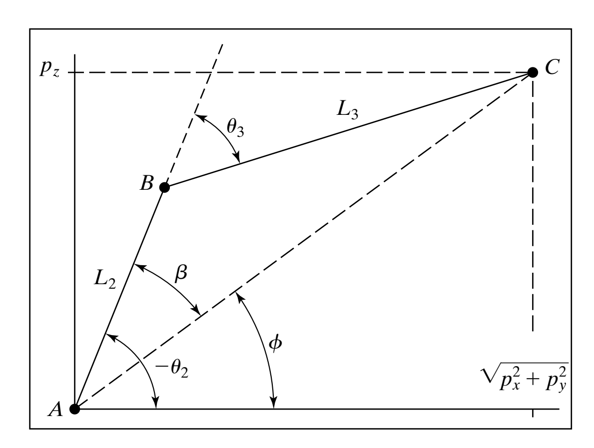


FIGURA 4.10: El plano del manipulador Motoman.

De la figura 4.10 podemos ver que $\theta_2 = -\phi - \beta$, o

$$\theta_2 = -\mathrm{Atan2}\left(p_z, \sqrt{p_x^2 + p_y^2}\right) - \mathrm{Atan2}(l_3 \operatorname{sen}\theta_3, l_2 + l_3 \cos\theta_3). \tag{4.103}$$

Finalmente, tenemos que

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3. \tag{4.104}$$

Después de resolver para los ángulos de articulación, debemos realizar el siguiente cálculo para obtener los valores de los actuadores. Si hacemos referencia a la sección 3.7 resolveremos la ecuación (3.16) para los A_i:

$$\begin{aligned}
A_1 &= \frac{1}{k_1}(\theta_1 - \lambda_1),\\
A_2 &= \frac{1}{k_2}\left(\sqrt{-2\alpha_2\beta_2 \cos\left(\theta_2 - \Omega_2 - \tan^{-1}\left(\frac{\phi_2}{\gamma_2}\right) + 270^\circ\right) + \alpha_2^2 + \beta_2^2} - \lambda_2\right)\\
A_3 &= \frac{1}{k_3}\left(\sqrt{-2\alpha_3\beta_3 \cos\left(\theta_2 + \theta_3 - \tan^{-1}\left(\frac{\phi_3}{\gamma_3}\right) + 90^\circ\right) + \alpha_3^2 + \beta_3^2} - \lambda_3\right),\\
A_4 &= \frac{1}{k_4}(180^\circ + \lambda_4 - \theta_2 - \theta_3 - \theta_4),\\
A_5 &= \frac{1}{k_5}(\lambda_5 - \theta_5).
\end{aligned} \tag{4.105}$$

Los actuadores tienen rangos limitados de movimiento, por lo que debemos comprobar que la solución que hayamos calculado esté dentro del rango. Esta comprobación "dentro del rango" se complica por el hecho de que el arreglo mecánico hace que los actuadores interactúen entre sí y afecten sus rangos de movimiento permitido. Para el robot Motoman, los actuadores 2 y 3 interactúan de una forma tal que debe obedecer siempre la siguiente relación:

$$A_2 - 10,000 > A_3 > A_2 + 3000. \tag{4.106}$$

Esto es, los límites del actuador 3 son una función de la posición del actuador 2. De manera similar,

$$32{,}000 - A_4 < A_5 < 55{,}000. \quad (4.107)$$

Ahora, una revolución de la articulación 5 corresponde a 25,600 conteos del actuador por lo que, cuando $A_4 > 2600$, hay dos posibles soluciones para A_5. Ésta es la única situación en la que el Yasukawa Motoman L-3 tiene más de una solución.

4.8 LAS TRAMAS ESTÁNDAR

La habilidad de despejar los ángulos de articulación es realmente el elemento central en muchos sistemas de control de robots. De nuevo, considere el paradigma indicado en la figura 4.11 que muestra las tramas estándar.

La manera en que se utilizan estas tramas en un sistema de robot general es la siguiente:

1. El usuario le indica al sistema en dónde se va a ubicar la trama estación. Ésta podría encontrarse en la esquina de una superficie de trabajo, como en la figura 4.12, o incluso sujetarse a una banda transportadora móvil. La trama estación $\{S\}$ se define en forma relativa a la trama base $\{B\}$.
2. El usuario especifica la descripción de la herramienta que está utilizando el robot y proporciona la especificación de la trama $\{T\}$. Cada herramienta que recoge el robot podría tener asociada una trama $\{T\}$ distinta. Observe que la misma herramienta sujeta de distintas formas requiere distintas definiciones de la trama $\{T\}$. $\{T\}$ se especifica de manera relativa a $\{W\}$; esto es, ${}^{W}_{T}T$.
3. El usuario especifica el punto de destino para el movimiento de un robot, proporcionando la descripción de la trama de destino $\{G\}$ relativa a la trama estación. A menudo las definiciones de $\{T\}$ y $\{S\}$ permanecen fijas para varios movimientos del robot. En este caso, una vez definidos, el usuario simplemente proporciona una serie de especificaciones de $\{G\}$.

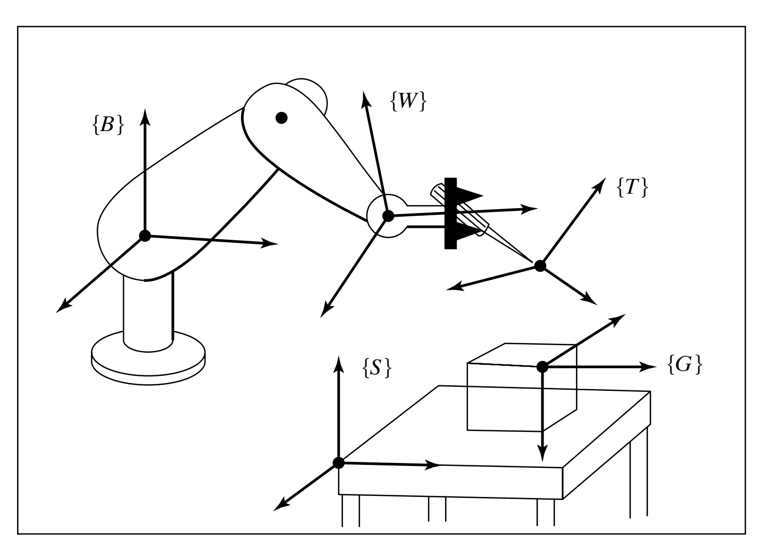


FIGURA 4.11: Ubicación de las tramas "estándar".

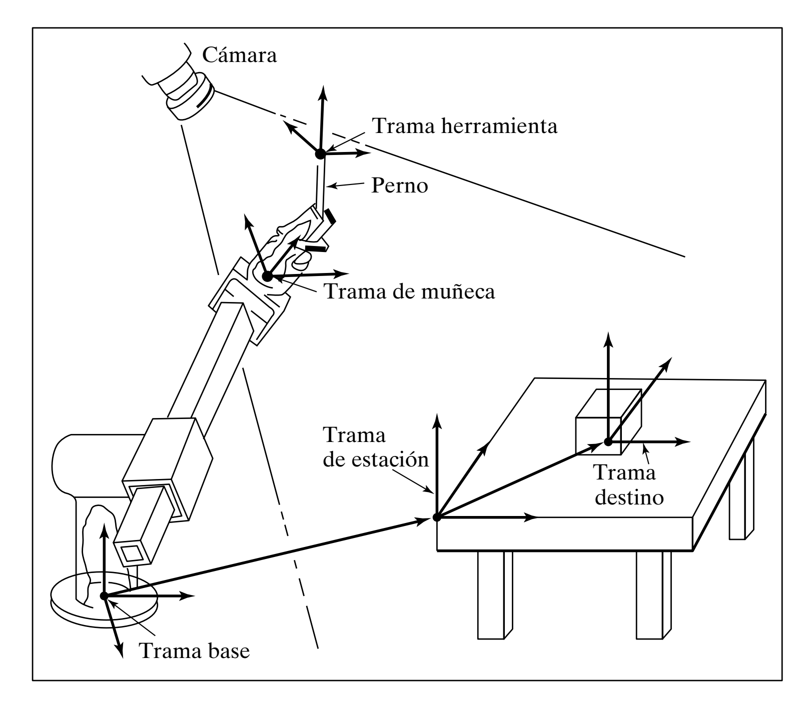


FIGURA 4.12: Estación de trabajo de ejemplo.

En muchos sistemas la definición de la trama herramienta (${}^{W}_{T}T$) es constante (por ejemplo, se define con su origen al centro de la punta de los dedos). Además, la trama estación podría ser fija o el usuario podría enseñarla fácilmente al mismo robot. En dichos sistemas el usuario no necesita conocer las cinco tramas estándar; simplemente piensa en términos de mover la herramienta a posiciones (destinos) respecto al área de trabajo especificada por la trama estación.

4. El sistema de robot calcula una serie de ángulos de articulación para mover las articulaciones de manera que la trama herramienta se mueva desde su posición inicial en forma uniforme hasta que $\{T\} = \{G\}$ al final del movimiento.

4.9 CÓMO RESOLVER UN MANIPULADOR

La función RESOLVER (SOLVE) implementa transformaciones cartesianas y llama a la función cinemática inversa, Por lo tanto, la cinemática inversa se generaliza de manera que puedan utilizarse definiciones arbitrarias para la trama herramienta y la trama estación con nuestra cinemática inversa, la cual despeja la trama de muñeca en términos de la trama base.

Dada la especificación de la trama de destino ${}^{S}_{T}T$, RESOLVER utiliza las definiciones de herramienta y estación para calcular la ubicación de $\{W\}$ relativa a $\{B\}$, ${}^{B}_{W}T$:

$$ {}^{B}_{W}T = {}^{B}_{S}T \; {}^{S}_{T}T \; {}^{W}_{T}T^{-1}. \tag{4.108}$$

Entonces, la cinemática inversa toma ${}^{B}_{W}T$ como entrada y calcula los valores de θ_1 a θ_n.

4.10 REPETITIVIDAD Y PRECISIÓN

En la actualidad muchos robots industriales se mueven hacia puntos de destino que les han sido enseñados. Un **punto enseñado** es uno al que el manipulador se mueve físicamente, después se leen los sensores de posición de articulación y se almacenan los ángulos de articulación. Cuando se ordena al robot que regrese a ese punto en el espacio, cada articulación se mueve al valor almacenado. En manipuladores simples del tipo "enseñar y reproducir" como éstos nunca surge el problema de cinemática inversa, ya que los puntos de destino nunca se especifican en coordenadas cartesianas. Cuando un fabricante especifica con qué precisión puede un manipulador regresar a un punto enseñado, está especificando la **repetitividad** del manipulador.

Siempre que se especifica una posición de destino y una orientación en términos cartesianos, debe calcularse la cinemática inversa del dispositivo para poder despejar los ángulos de articulación requeridos. Los sistemas que permiten describir destinos en términos cartesianos son capaces de mover el manipulador a puntos que nunca fueron enseñados; puntos en el espacio de trabajo a los cuales tal vez nunca haya ido antes. A esos puntos los llamaremos **puntos calculados**. Dicha capacidad es necesaria para muchas tareas de manipulación. Por ejemplo, si se utiliza un sistema de visión computacional para localizar una pieza que el robot debe sujetar, éste debe poder moverse a las coordenadas cartesianas proporcionadas por el sensor de visión. La precisión con la que se puede llegar a un punto calculado se conoce como **precisión** del manipulador.

La precisión de un manipulador está confinada por la repetitividad. Evidentemente la precisión se ve afectada por la precisión de los parámetros que aparecen en las ecuaciones cinemáticas del robot. Si no se conocen bien los parámetros Denavit-Hartenberg las ecuaciones cinemáticas inversas calcularán valores de ángulos de articulación erróneos. Por ende, aunque la repetitividad de la mayoría de los manipuladores industriales es bastante buena, la precisión es generalmente mucho peor y varía considerablemente de un manipulador a otro. Pueden idearse técnicas de calibración que permitan mejorar la precisión de un manipulador a través de la estimación de esos parámetros cinemáticos específicos del manipulador [10].

4.11 CONSIDERACIONES COMPUTACIONALES

En muchos esquemas de control de trayectoria, los cuales consideraremos en el capítulo 7, es necesario calcular la cinemática inversa de un manipulador a frecuencias considerablemente altas; por ejemplo de 30Hz o más. Por lo tanto, la eficiencia computacional es una cuestión importante a considerar. Estos requerimientos de velocidad descartan el uso de técnicas de solución numérica que son iterativas por naturaleza y que, por esta razón, no hemos considerado.

La mayor parte de los comentarios generales de la sección 3.10, dirigidos a la cinemática directa, también se aplican para el problema de la cinemática inversa. Para este caso se utiliza comúnmente una rutina Atan2 de búsqueda en tabla para obtener velocidades mayores.

También es importante la estructura de la computación de múltiples soluciones. Generalmente es más eficiente evaluarlas en paralelo, en vez de hacerlo en forma serial. Desde luego que, en ciertas aplicaciones en donde no se requieren todas las soluciones, ahorra mucho tiempo.

Cuando se utiliza un método geométrico para desarrollar una solución cinemática inversa, en ocasiones es posible calcular varias soluciones mediante operaciones simples sobre los diversos ángulos que se despejaron al obtener la primera solución. Esto

es, la primera solución es moderadamente costosa en términos computacionales, pero las demás soluciones se encuentran con mucha rapidez al sumar y diferenciar ángulos, restar π, etcétera.

BIBLIOGRAFÍA

[1] B. Roth, J. Rastegar y V. Scheinman, "On the Design of Computer Controlled Manipulators", *On the Theory and Practice of Robots and Manipulators*, Vol. 1, Primer Simposium CISM-IFToMM, septiembre de 1973, pp. 93-113.

[2] B. Roth, "Performance Evaluation of Manipulators from a Kinematic Viewpoint", *Performance Evaluation of Manipulators*, Agencia Nacional de Estándares, publicación especial, 1975.

[3] D. Pieper y B. Roth, *The Kinematics of Manipulators Under Computer Control*, *Proceedings of the Second International Congress on Theory of Machines and Mechanisms*, Vol. 2, Zakopane, Polonia, 1969, pp. 159-169.

[4] D. Pieper, "The Kinematics of Manipulators Under Computer Control", Tesis doctoral sin publicar, Universidad de Stanford, 1968.

[5] R.P. Paul, B. Shimano y G. Mayer, "Kinematic Control Equations for Simple Manipulators", *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. SMC-11, Núm. 6, 1981.

[6] L. Tsai y A. Morgan, "Solving the Kinematics of the Most General Six- and Five-degree-of-freedom Manipulatos by Continuation Methods", artículo 84-DET-20, Conferencia de Mecanismos de la ASME, Boston, Octubre 7-10, 1984.

[7] C.S.G. Lee y M. Ziegler, "Geometric Approach in Solving Inverse Kinematics of PUMA Robots", *IEEE Transactions on Aerospace and Electronics Systems*, Vol. AES-20, Núm. 6, noviembre de 1984.

[8] W. Beyer, *CRC Standard Mathematical Tables*, 25a. edición, CRC Press, Inc., Boca Raton, FL, 1980.

[9] R. Burington, *Handbook of Mathematical Tables and Formulas*, 5a. Edición, McGraw-Hill, Nueva York, 1973.

[10] J. Hollerbach, "A Survey of Kinematic Calibration", en *The Robotics Review*, O. Khatib, J. Craig y T. Lozano-Perez, Editores, MIT Press, Cambridge, MA, 1989.

[11] Y. Nakamura y H. Hanafusa, "Inverse Kinematic Solutions with Singularity Robustness for Robot Manipulator Control", *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 108, 1986.

[12] D. Baker and C. Wampler, "On the Inverse Kinematics of Redundant Manipulators", *International Journal of Robotics Research*, Vol. 7, Núm. 2, 1988.

[13] L.W. Tsai, *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*, Wiley, Nueva York, 1999.

EJERCICIOS

4.1 [15] Bosqueje el espacio de trabajo de la punta de los dedos del manipulador de tres vínculos del capítulo 3, ejercicio 3.3 para el caso $l_1 = 15.0$, $l_2 = 10.0$ y $l_3 = 3.0$.

4.2 [26] Derive la cinemática inversa del manipulador de tres vínculos del capítulo 3, ejercicio 3.3.

4.3 [12] Bosqueje el espacio de trabajo de la punta de los dedos del manipulador 3-DOF del capítulo 3, ejemplo 3.4.

4.4 [24] Derive la cinemática inversa del manipulador 3-DOF del capítulo 3, ejemplo 3.4.

4.5 [38] Escriba una subrutina en Pascal (o C) que calcule todas las posibles soluciones para el manipulador PUMA 560 que se encuentren dentro de los siguientes límites de articulación:

$$-170.0 < \theta_1 < 170.0,$$
$$-225.0 < \theta_2 < 45.0,$$
$$-250.0 < \theta_3 < 75.0,$$
$$-135.0 < \theta_4 < 135.0,$$
$$-100.0 < \theta_5 < 100.0,$$
$$-180.0 < \theta_6 < 180.0.$$

Use las ecuaciones derivadas en la sección 4.7 con estos valores numéricos (en pulgadas):

$$a_2 = 17.0,$$
$$a_3 = 0.8,$$
$$d_3 = 4.9,$$
$$d_4 = 17.0.$$

4.6 [15] Describa un algoritmo simple para seleccionar la solución más cercana de un conjunto de posibles soluciones.

4.7 [10] Haga una lista de factores que podrían afectar la repetitividad de un manipulador. Haga una segunda lista de factores adicionales que afectan la precisión de un manipulador.

4.8 [12] Dada una posición y orientación deseadas de la mano de un manipulador planar de articulaciones giratorias y tres vínculos, hay dos posibles soluciones. Si agregamos una articulación giratoria más (de manera tal que el brazo siga siendo planar), ¿cuántas soluciones habrá?

4.9 [26] La figura 4.13 muestra un brazo planar de dos vínculos con articulaciones giratorias. Para este brazo el segundo vínculo es la mitad de largo que el primero; esto es, $l_1 = 2l_2$. Los límites de rango de las articulaciones en grados son

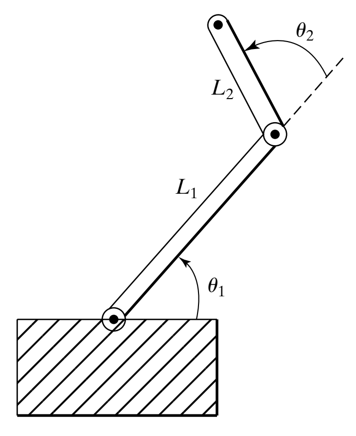


FIGURA 4.13: Manipulador planar de dos vínculos.

$$0 < \theta_1 < 180,$$
$$-90 < \theta_2 < 180.$$

Bosqueje el espacio de trabajo alcanzable aproximado (un área) de la punta del vínculo 2.

4.10 [23] Proporcione una expresión para el subespacio del manipulador del capítulo 3, ejemplo 3.4.

4.11 [24] Una tabla de posicionamiento 2-DOF se utiliza para orientar piezas para soldadura de arco. La cinemática directa que ubica la cubierta de la mesa (vínculo 2) con respecto a la base (vínculo 0) es

$$ {}^0_2T = \begin{bmatrix} c_1c_2 & -c_1s_2 & s_1 & l_2s_1 + l_1 \\ s_2 & c_2 & 0 & 0 \\ -s_1c_2 & s_1s_2 & c_1 & l_2c_1 + h_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

Dada cualquier dirección unitaria fija en la trama de la cubierta (vínculo 2) representada por ${}^2\hat{V}$, proporcione la solución de cinemática inversa para θ_1, θ_2 de manera tal que este vector esté alineado con ${}^0\hat{Z}$ (es decir, hacia arriba). ¿Hay múltiples soluciones? ¿Existe una sola condición para la que no pueda obtenerse una solución única?

4.12 [22] En la figura 4.14 se ilustran dos mecanismos $3R$. En ambos casos los tres ejes se intersecan en un punto (y, sobre todas las configuraciones, este punto permanece fijo en el espacio). El mecanismo de la figura 4.14(a) tiene un ángulo de torsión de vínculo (α_i) con magnitud de 90 grados. El mecanismo de la figura 4.14(b) tiene una torsión de magnitud ϕ y otra de $180 - \phi$ en magnitud.

Se puede ver que el mecanismo de la figura 4.14(a) se encuentra en correspondencia con los ángulos de Euler Z-Y-Z, y por lo tanto sabemos que basta con orientar el vínculo 3 (con la flecha en la figura) arbitrariamente con respecto al vínculo 0 fijo. Como ϕ no es igual a 90 grados, resulta que el otro mecanismo no puede orientar el vínculo 3 de manera arbitraria.

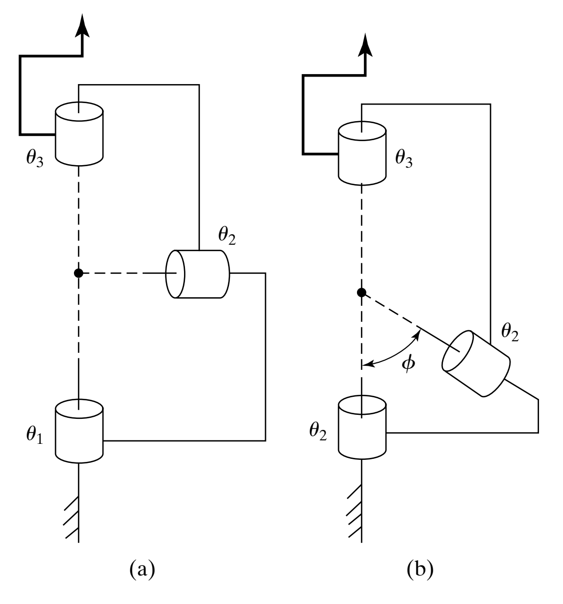


FIGURA 4.14: Dos mecanismos $3R$ (ejercicio 4.12).

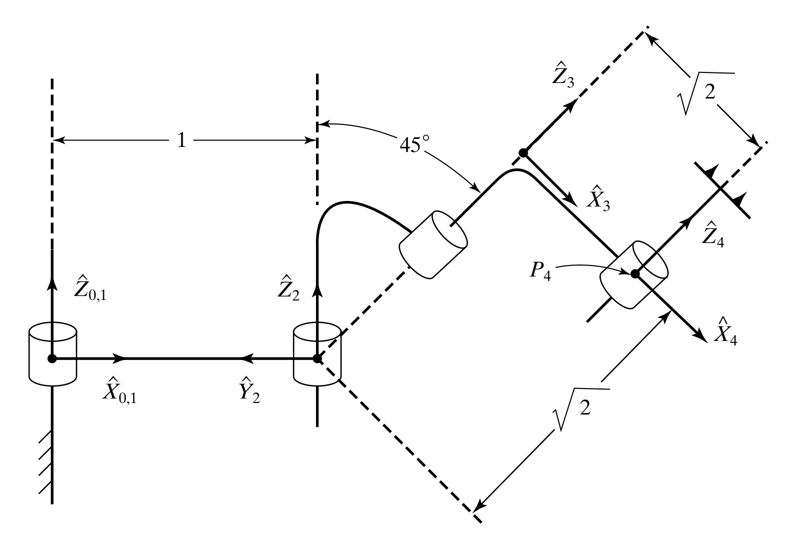


FIGURA 4.15: Un manipulador $4R$ mostrado en la posición $\Theta = [0, 90°, -90°, 0]T$ (ejercicio 4.16).

Describa el conjunto de orientaciones que son *inalcanzables* con el segundo mecanismo. Observe que asumimos que todas las articulaciones pueden girar 360 grados (es decir, sin límites) y suponemos que los vínculos pueden pasar uno a través del otro si es necesario (es decir, el espacio de trabajo no está limitado por autocolisiones).

4.13 [13] Mencione dos razones para las que se prefieren soluciones cinemáticas analíticas de forma cerrada en vez de soluciones iterativas.

4.14 [14] Existen robots 6-DOF para los que la cinemática NO puede resolverse en forma cerrada. ¿Existe algún robot 3-DOF para el que la cinemática (de posición) NO pueda resolverse en forma cerrada?

4.15 [38] Escriba una subrutina que resuelva ecuaciones cuadráticas en forma cerrada. (Vea [8, 9]).

4.16 [25] Un manipulador $4R$ se muestra esquemáticamente en la figura 4.15. Los parámetros de vínculo distintos de cero son $a_1 = 1$, $\alpha_2 = 45°$, $d_3 = \sqrt{2}$ y $a_3 = \sqrt{2}$, y el mecanismo se ilustra en la configuración que corresponde a $\Theta = [0, 90°, -90°, 0]^T$. Cada articulación tiene $\pm 180°$ como límites. Encuentre todos los valores de θ_2 tales que

$$^0P_{4ORG} = [1.1, 1.5, 1.707]^T.$$

4.17 [25] La figura 4.16 muestra esquemáticamente un manipulador *4R*. Los parámetros de vínculo distintos de cero son $\alpha_1 = -90°$, $d_2 = 1$, $\alpha_2 = 45°$, $d_3 = 1$ y $a_3 = 1$, y el mecanismo se ilustra en la configuración que corresponde a $\Theta = [0, 0, 90°, 0]^T$. Cada articulación tiene $\pm 180°$ como límites. Encuentre todos los valores de θ_3 tales que

$$^0P_{4ORG} = [0.0, 1.0, 1.414]^T.$$

4.18 [15] Considere el manipulador *RRP* que se muestra en la figura 3.37. ¿Cuántas soluciones (posición) poseen las ecuaciones cinemáticas?

4.19 [15] Considere el manipulador *RRR* que se muestra en la figura 3.38. ¿Cuántas soluciones (posición) poseen las ecuaciones cinemáticas?

4.20 [15] Considere el manipulador *RPP* que se muestra en la figura 3.39. ¿Cuántas soluciones (posición) poseen las ecuaciones cinemáticas?

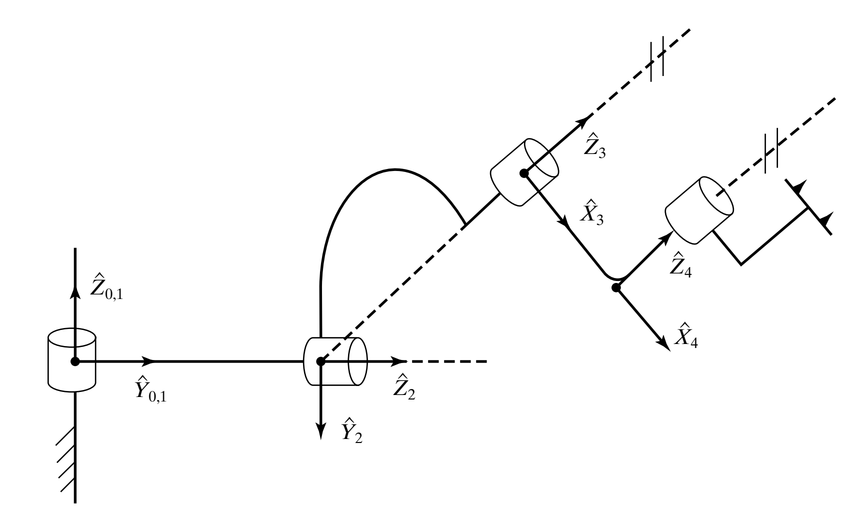


FIGURA 4.16: Un manipulador 4*R* en la posición $\Theta = [0, 0, 90°, 0]^T$ (ejercicio 4.17).

4.21 [15] Considere el manipulador *PRR* que se muestra en la figura 3.40. ¿Cuántas soluciones (posición) poseen las ecuaciones cinemáticas?

4.22 [15] Considere el manipulador *PPP* que se muestra en la figura 3.41. ¿Cuántas soluciones (posición) poseen las ecuaciones cinemáticas?

4.23 [38] Las siguientes ecuaciones cinemáticas surgen en cierto problema:

$$\begin{aligned} \operatorname{sen}\xi &= a \operatorname{sen}\theta + b, \\ \operatorname{sen}\phi &= c\cos\theta + d, \\ \psi &= \xi + \phi. \end{aligned}$$

Dadas a, b, c, d y ψ muestre que, en el caso general, existen cuatro soluciones para θ. Proporcione una condición especial bajo la cual haya solamente dos soluciones para θ.

4.24 [20] Dada la descripción de la trama de vínculo $\{i\}$ en términos de la trama de vínculo $\{i-1\}$, encuentre los cuatro parámetros de Denavit-Hartenberg como funciones de los elementos de ${}^{i-1}_{i}T$.

EJERCICIO DE PROGRAMACIÓN (PARTE 4)

1. Escriba una subrutina para calcular la cinemática inversa para el manipulador de tres vínculos de la sección 4.4. La rutina debe pasar argumentos en la siguiente forma:

```
Procedure CININV(VAR wrelb: trama; VAR actual, cerca, lejos: vec3;
VAR sol: boolean);
```

en donde “wrelb” (entrada) es la trama de muñeca especificada en forma relativa a la trama base ; “actual” (entrada) es la posición actual del robot (dada como un vector de ángulos de articulación); “cerca” es la solución más cercana; “lejos” es la segunda solución, y “sol” es una bandera que indica si se encontraron soluciones o no. (sol = FALSE si no se encontraron soluciones). Las longitudes de vínculo (en metros) son:

$$l_1 = l_2 = 0.5.$$

Los rangos de movimiento de las articulaciones son:

$$-170° \leq \theta_i \leq 170°.$$

Pruebe su rutina llamándola en conjunto con CIN para demostrar que son ciertamente inversas una de la otra.

2. Hay una herramienta unida al vínculo 3 del manipulador. Esta herramienta está descrita por ${}^{W}_{T}T$, la trama herramienta relativa a la trama de la muñeca. Además, un usuario ha descrito su área de trabajo, la trama estación relativa a la base del robot, como ${}^{B}_{S}T$. Escriba la subrutina

```
Procedure RESOLVER(VAR trels: trama; VAR actual, cerca,
lejos: vec3; VAR sol: boolean);
```

en donde "trels" es la trama $\{T\}$ especificada en forma relativa a la trama $\{S\}$. Los demás parámetros son exactamente iguales a los de la subrutina CININV. Las definiciones de $\{T\}$ y $\{S\}$ deben ser variables o constantes definidas en forma global. RESOLVER debe utilizar llamadas a `TMULT, TINVERT` y `CININV`.

3. Escriba la secuencia principal del programa para que acepte una trama destino especificada en términos de x, y y ϕ. Esta especificación de destino es $\{T\}$ relativa a $\{S\}$, que es la manera en que el usuario desea especificar destinos.
El robot utiliza la misma herramienta en la misma área de trabajo que en el ejercicio de programación (parte 2), por lo que $\{T\}$ y $\{S\}$ se definen como

$$ {}^{W}_{T}T = [x\ y\ \theta] = [0.1\ 0.2\ 30.0], $$

$$ {}^{B}_{S}T = [x\ y\ \theta] = [-0.1\ 0.3\ 0.0]. $$

Calcule los ángulos de articulación para cada una de las siguientes tramas de destino:

$$[x_1\ y_1\ \phi_1] = [0.0\ 0.0\ -90.0],$$

$$[x_2\ y_2\ \phi_2] = [0.6\ -0.3\ 45.0],$$

$$[x_3\ y_3\ \phi_3] = [-0.4\ 0.3\ 120.0],$$

$$[x_4\ y_4\ \phi_4] = [0.8\ 1.4\ 30.0].$$

Suponga que el robot empezará con todos los ángulos iguales a 0.0 y se moverá a estos tres destinos en secuencia. El programa deberá encontrar la solución más cercana respecto al punto de destino anterior. Usted deberá llamar consecutivamente a `RESOLVER` y `DONDE` para asegurarse que sean verdaderamente funciones inversas.

EJERCICIO MATLAB 4

Este ejercicio se enfoca en la solución de cinemática de planteamiento inverso para el robot planar 3-DOF, 3R. (Vea las figuras 3.6 y 3.7; los parámetros de DH se proporcionan en la figura 3.8). Se proporcionan los siguientes parámetros de longitud fija: $L_1 = 4$, $L_2 = 3$ y $L_3 = 2$(m).

a) Derive en forma analítica y a mano, la solución de planteamiento inverso para este robot. Dada ${}^{0}_{H}T$, calcule todas las múltiples soluciones posibles para $\{\theta_1\ \theta_2\ \theta_3\}$. (Se presentan tres métodos en el texto; elija uno de ellos). Sugerencia: para simplificar las ecuaciones, primero calcule ${}^{0}_{3}T$ a partir de ${}^{0}_{H}T$ y L_3.

b) Desarrolle un programa de MATLAB para resolver por completo este problema de cinemática de planteamiento inverso para el robot 3R planar (es decir, proporcione todas las múltiples soluciones). Pruebe su programa utilizando los siguientes casos de entrada:

i) $${}^0_HT = \begin{bmatrix} 1 & 0 & 0 & 9 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

ii) $${}^0_HT = \begin{bmatrix} 0.5 & -0.866 & 0 & 7.5373 \\ 0.866 & 0.6 & 0 & 3.9266 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

iii) $${}^0_HT = \begin{bmatrix} 0 & 1 & 0 & -3 \\ -1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

iv) $${}^0_HT = \begin{bmatrix} 0.866 & 0.5 & 0 & -3.1245 \\ -0.5 & 0.866 & 0 & 9.1674 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Para todos los casos emplee una comprobación circular para validar sus resultados: introduzca cada conjunto de ángulos de articulación (para cada una de las múltiples soluciones) de vuelta en el programa MATLAB de cinemática de planteamiento directo para demostrar que obtiene la matriz ${}^0_HT$ original.

c) Compruebe todos los resultados por medio de la Caja de Herramientas de Robótica MATLAB de Corke. Pruebe la función *ikine()*.

C A P Í T U L O 5

Jacobianos: velocidades y fuerzas estáticas

5.1 INTRODUCCIÓN

En este capítulo expandiremos nuestra consideración de los manipuladores de robots más allá de los problemas de posicionamiento estático. Examinaremos las nociones de velocidad lineal y angular de un cuerpo rígido y utilizaremos estos conceptos para analizar el movimiento de un manipulador. También consideraremos las fuerzas que actúan sobre un cuerpo rígido y después utilizaremos estas ideas para estudiar la aplicación de fuerzas estáticas con los manipuladores.

Resulta ser que el estudio de ambas velocidades y fuerzas estáticas nos lleva a una matriz llamada el **jacobiano**[1] del manipulador, que presentaremos en este capítulo.

El campo de la cinemática de mecanismos no se trata aquí con gran profundidad. En su mayor parte la presentación se restringe sólo a aquellas ideas que son fundamentales para el problema específico de la robótica. Si desea profundizar en este tema consulte algún texto sobre mecánica [1 a 3].

5.2 NOTACIÓN PARA LA POSICIÓN Y ORIENTACIÓN VARIANTES EN EL TIEMPO

Antes de investigar la descripción del movimiento de un cuerpo rígido, hablaremos brevemente sobre ciertas cuestiones básicas: la diferenciación de vectores, la representación de la velocidad angular y la notación.

[1] Los matemáticos la llaman a veces "matriz jacobiana", pero los robóticos generalmente la abrevian simplemente como "jacobiano".

Diferenciación de los vectores de posición

Como base para nuestra consideración de velocidades (y, en el capítulo 6, de aceleraciones) necesitaremos la siguiente notación para la derivada de un vector:

$$^{B}V_Q = \frac{d}{dt}\,{}^{B}Q = \lim_{\Delta t \to 0} \frac{^{B}Q(t+\Delta t) - {}^{B}Q(t)}{\Delta t}. \tag{5.1}$$

La velocidad de un vector de posición puede definirse como la velocidad lineal del punto en el espacio representado por el vector de posición. De la ecuación (5.1), podemos ver que estamos calculando la derivada de Q relativa a la trama $\{B\}$. Por ejemplo, si Q no está cambiando en el tiempo respecto a $\{B\}$, entonces la velocidad calculada es cero; aun si hay alguna otra trama en la que Q esté variando. Por lo tanto, es importante indicar la trama en la que se diferencia el vector.

Al igual que con cualquier vector, un vector de velocidad puede describirse en términos de cualquier trama, y esta trama de referencia se indica con un superíndice a la izquierda. Por ende, el vector de velocidad calculado por la ecuación (5.1) expresado en términos de la trama $\{A\}$ se escribiría así:

$$^{A}(^{B}V_Q) = \frac{^{A}d}{dt}\,{}^{B}Q. \tag{5.2}$$

Entonces vemos que, en el caso general, un vector de velocidad está asociado con un punto en el espacio, pero los valores numéricos que describen la velocidad de ese punto dependen de dos tramas: una respecto a la que se realizó la diferenciación y otra respecto a la que se expresó el vector de velocidad resultante.

En la ecuación (5.1), la velocidad calculada se escribe en términos de la trama de diferenciación, por lo que el resultado podría indicarse con un superíndice B a la izquierda pero, por simplicidad, cuando ambos superíndices son iguales, no necesitamos indicar el exterior; es decir, podemos escribir:

$$^{B}(^{B}V_Q) = {}^{B}V_Q. \tag{5.3}$$

Por último, podemos eliminar el superíndice externo a la izquierda, incluyendo explícitamente la matriz de rotación que realiza el cambio en la trama de referencia (vea la sección 2.10); esto es, escribimos lo siguiente:

$$^{A}(^{B}V_Q) = {}^{A}_{B}R\,{}^{B}V_Q. \tag{5.4}$$

En general escribiremos expresiones de la forma del lado derecho de la ecuación (5.4), de manera que los símbolos que representan velocidades siempre indicarán la velocidad en la trama de diferenciación y no tendrán superíndices externos a la izquierda.

En vez de considerar la velocidad de un punto general relativa a una trama arbitraria, muy a menudo consideraremos la velocidad del *origen de una trama* relativo a alguna trama de referencia universal que pueda comprenderse. Para este caso especial definiremos una notación abreviada,

$$v_C = {}^{U}V_{CORG}, \tag{5.5}$$

en donde el punto en cuestión es el origen de la trama $\{C\}$ y la trama de referencia es $\{U\}$. Por ejemplo, podemos usar la notación v_C para referirnos a la velocidad del origen de la trama $\{C\}$; entonces $^{A}v_C$ es la velocidad del origen de la trama $\{C\}$ expresada en términos de la trama $\{A\}$ (aunque la diferenciación se hizo con referencia a $\{U\}$).

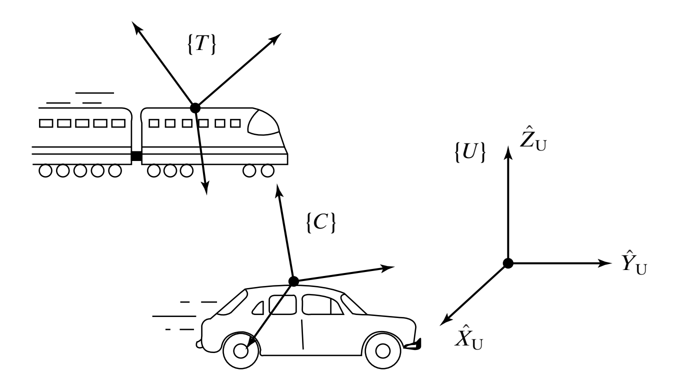


FIGURA 5.1: Ejemplo de algunas tramas en movimiento lineal.

EJEMPLO 5.1

La figura 5.1 muestra una trama universal fija $\{U\}$, una trama $\{T\}$ unida a un tren que viaja a 100 mph y una trama $\{C\}$ unida a un auto que viaja a 30 mph. Ambos vehículos se dirigen en la misma dirección $\hat{X}$ de $\{U\}$. Las matrices de rotación ${}^U_TR$ y ${}^U_CR$ se conocen y son constantes.

¿Cuál es el valor de $\frac{{}^Ud}{dt}\,{}^UP_{CORG}$?

$$\frac{{}^Ud}{dt}\,{}^UP_{CORG} = {}^UV_{CORG} = \upsilon_C = 30\hat{X}.$$

¿Cuál es el valor de ${}^C({}^UV_{TORG})$?

$${}^C({}^UV_{TORG}) = {}^C\upsilon_T = {}^C_UR\upsilon_T = {}^C_U\,R(100\hat{X}) = {}^U_CR^{-1}\,100\hat{X}.$$

¿Cuál es el valor de ${}^C({}^TV_{CORG})$?

$${}^C({}^TV_{CORG}) = {}^C_TR\;{}^TV_{CORG} = -{}^U_CR^{-1}\,{}^U_TR\;70\hat{X}.$$

El vector de velocidad angular

Ahora presentaremos un **vector de velocidad angular** que utiliza el símbolo Ω. Mientras que la velocidad lineal describe un atributo de un punto, la velocidad angular describe un atributo de un cuerpo. Siempre unimos una trama a los cuerpos que consideramos, por lo que también podemos pensar que la velocidad angular describe el movimiento giratorio de una trama.

En la figura 5.2, ${}^A\Omega_B$ describe la rotación de la trama $\{B\}$ relativa a la trama $\{A\}$. Físicamente, en cualquier instante la dirección de ${}^A\Omega_B$ indica el eje instantáneo de rotación de $\{B\}$ relativo a $\{A\}$, y la magnitud de ${}^A\Omega_B$ indica la velocidad de rotación. De nuevo y como en cualquier vector, un vector de velocidad angular puede expresarse en cualquier sistema de coordenadas, por lo que puede agregarse otro superíndice a la izquierda; por ejemplo, ${}^C({}^A\Omega_B)$ es la velocidad angular de la trama $\{B\}$ relativa a $\{A\}$, expresada en términos de la trama $\{C\}$.

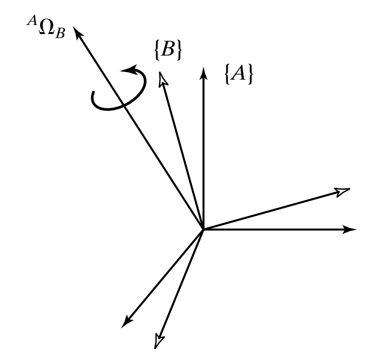


FIGURA 5.2: La trama $\{B\}$ gira con la velocidad angular $^A\Omega_B$ relativa a la trama $\{A\}$.

De nuevo, introduciremos una notación simplificada para un caso especial importante. Éste es simplemente el caso en el que hay una trama de referencia conocida, de manera que no necesita mencionarse en la notación:

$$\omega_C = {}^U\Omega_C. \tag{5.6}$$

Aquí ω_C es la velocidad angular de la trama $\{C\}$ relativa a cierta trama de referencia conocida $\{U\}$. Por ejemplo, $^A\omega_C$ es la velocidad angular de la trama $\{C\}$ expresada en términos de $\{A\}$ (aunque la velocidad angular es con respecto a $\{U\}$).

5.3 VELOCIDAD LINEAL Y ROTACIONAL DE CUERPOS RÍGIDOS

En esta sección, investigaremos la descripción del movimiento de un cuerpo rígido, cuando menos en lo que respecta a la velocidad. Estas ideas extienden las nociones de traslaciones y orientaciones descritas en el capítulo 2 al caso en el que hay variación con el tiempo. En el capítulo 6 extenderemos aún más nuestro estudio para considerar la aceleración.

Al igual que en el capítulo 2, unimos un sistema de coordenadas a cualquier cuerpo que deseemos describir. De esta manera, el movimiento de los cuerpos rígidos puede estudiarse de manera equivalente al movimiento de las tramas relativas entre sí.

Velocidad lineal

Considere una trama $\{B\}$ unida a un cuerpo rígido. Deseamos describir el movimiento de ^BQ respecto a la trama $\{A\}$, como se muestra en la figura 5.3. Podemos considerar que $\{A\}$ es fija.

La trama $\{B\}$ se ubica de manera relativa a $\{A\}$ como lo describe un vector de posición $^AP_{BORG}$ y una matriz de rotación ^A_BR. De momento, supondremos que la orientación ^A_BR no cambia con el tiempo; esto es, el movimiento del punto Q respecto a $\{A\}$ se debe a que $^AP_{BORG}$ o ^BQ cambian con el tiempo.

Es bastante simple resolver para la velocidad lineal del punto Q en términos de $\{A\}$. Sólo se expresan ambos componentes de la velocidad en términos de $\{A\}$ y se suman:

$$^AV_Q = {}^AV_{BORG} + {}^A_BR\, {}^BV_Q. \tag{5.7}$$

La ecuación (5.7) es solamente para el caso en el que la orientación relativa de $\{B\}$ y $\{A\}$ permanece constante.

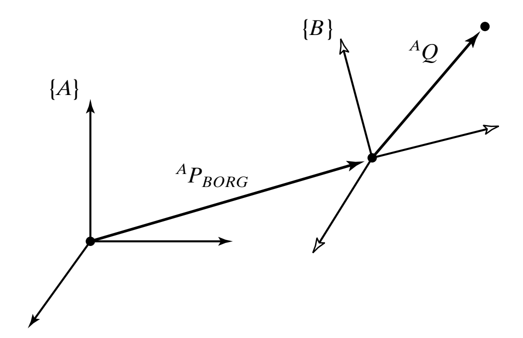


FIGURA 5.3: La trama $\{B\}$ se traslada con la velocidad $^{A}V_{BORG}$ relativa a la trama $\{A\}$.

Velocidad de rotación

Ahora consideremos dos tramas con orígenes coincidentes y velocidad relativa lineal cero; sus orígenes permanecerán coincidentes en todo momento. Una o ambas podrían estar fijas a cuerpos rígidos pero, por claridad, los cuerpos rígidos no se muestran en la figura 5.4.

La orientación de la trama $\{B\}$ con respecto a la trama $\{A\}$ está cambiando en el tiempo. Como se indica en la figura 5.4, la velocidad de rotación de $\{B\}$ relativa a $\{A\}$ es descrita por un vector llamado $^{A}\Omega_{B}$. También hemos indicado un vector ^{B}Q que localiza un punto fijo en $\{B\}$. Ahora consideremos la pregunta más importante: ¿cómo cambia un vector con el tiempo según se ve desde $\{A\}$ cuando está fijo en $\{B\}$ y los sistemas están rotando?

Consideremos que el vector Q es constante viéndolo desde la trama {B}; esto es,

$$^{B}V_{Q} = 0. \tag{5.8}$$

Aunque es constante de manera relativa a {B}, está claro que el punto Q tendrá una velocidad como se ve desde {A} debido a la velocidad rotacional $^{A}\Omega_{B.}$ Para conocer la velocidad del punto Q, usaremos una aproximación intuitiva. La figura 5.5 muestra dos fracciones de tiempo, mientras el vector Q gira alrededor de $^{A}\Omega_{B}$. Esto es lo que un observador vería en {A}.

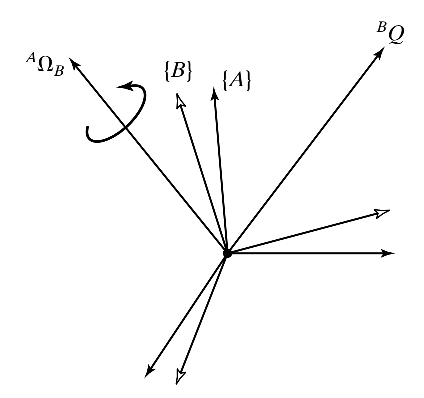


FIGURA 5.4: El vector ^{B}Q fijo en la trama $\{B\}$ está girando respecto a la trama $\{A\}$ con una velocidad angular $^{A}\Omega_{B}$.

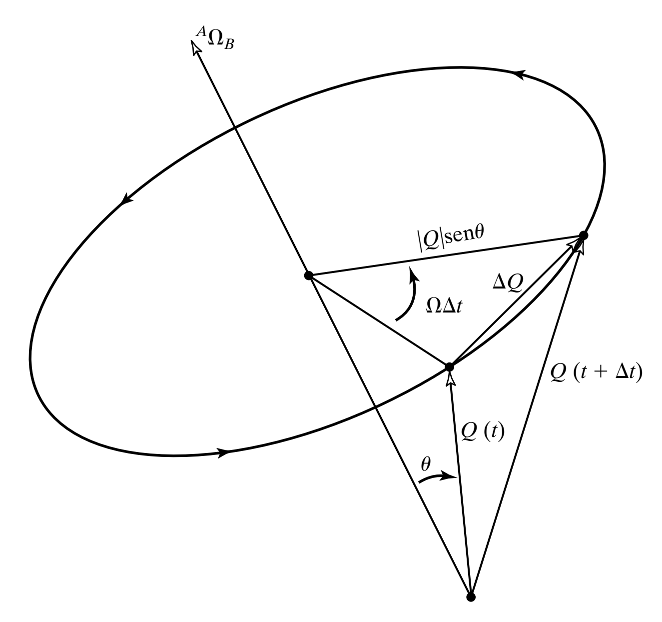


FIGURA 5.5: La velocidad de un punto debido a una velocidad angular.

Al examinar la figura 5.5 podemos averiguar tanto la dirección como la magnitud del cambio en el vector, visto desde $\{A\}$. En primer lugar, está claro que el cambio diferencial en ^AQ debe ser perpendicular tanto a $^A\Omega_B$ como a ^AQ. En segundo lugar, podemos ver gracias a la figura 5.5 que la magnitud del cambio diferencial es

$$|\Delta Q| = (|^A Q| \operatorname{sen} \theta)(|^A \Omega_B| \Delta t). \tag{5.9}$$

Estas condiciones sobre magnitud y dirección sugieren inmediatamente el producto cruz de los vectores. Evidentemente, nuestras conclusiones sobre dirección y magnitud se satisfacen mediante la siguiente forma computacional:

$$^A V_Q = {}^A\Omega_B \times {}^A Q. \tag{5.10}$$

En el caso general, el vector Q también podría estar cambiando respecto a la trama $\{B\}$, por lo que si sumamos este componente tenemos que:

$$^A V_Q = {}^A({}^B V_Q) + {}^A\Omega_B \times {}^A Q. \tag{5.11}$$

Si utilizamos una matriz de rotación para eliminar el doble superíndice y, teniendo en cuenta que la descripción de ^AQ en cualquier instante es $^A_BR^BQ$, terminamos con

$$^A V_Q = {}^A_B R\, {}^B V_Q + {}^A\Omega_B \times {}^A_B R\, {}^B Q. \tag{5.12}$$

Velocidad lineal y de rotación simultáneas

Podemos expandir de manera muy simple la ecuación (5.12) para el caso en el que los orígenes no son coincidentes; sumando la velocidad lineal del origen a la ecuación (5.12) para derivar la fórmula general para la velocidad de un vector fijo en la trama $\{B\}$, visto desde la trama $\{A\}$:

$$^A V_Q = {}^A V_{BORG} + {}^A_B R\, {}^B V_Q + {}^A\Omega_B \times {}^A_B R\, {}^B Q \tag{5.13}$$

La ecuación (5.13) es el resultado final para la derivada de un vector en una trama móvil, visto desde una trama estacionaria.

5.4 MÁS SOBRE LA VELOCIDAD ANGULAR

En esta sección daremos un vistazo más detallado a la velocidad angular y, en especial, a la derivación de (5.10). Mientras que en la sección anterior tomamos un método geométrico para demostrar la validez de la ecuación (5.10), aquí lo haremos con uno matemático. Si es la primera vez que lee este libro, puede omitir esta sección.

Una propiedad de la derivada de una matriz ortonormal

Podemos obtener una interesante relación entre la derivada de una matriz ortonormal y cierta matriz antisimétrica como se muestra a continuación. Para cualquier matriz ortonormal de $n \times n$ de nombre R, tenemos que

$$RR^T = I_n \tag{5.14}$$

en donde I_n es la matriz de identidad de $n \times n$. Por cierto, a nosotros nos interesa el caso en el que $n = 3$ y R es una matriz ortonormal *propia*, o matriz de rotación. Si diferenciamos la ecuación (5.14) nos queda

$$\dot{R}R^T + R\dot{R}^T = 0_n, \tag{5.15}$$

en donde 0_n es la matriz cero de $n \times n$. La ecuación (5.15) también puede escribirse así:

$$\dot{R}R^T + (\dot{R}R^T)^T = 0_n. \tag{5.16}$$

Definiendo

$$S = \dot{R}R^T, \tag{5.17}$$

de la ecuación (5.16) tenemos que

$$S + S^T = 0_n. \tag{5.18}$$

Por lo tanto, podemos ver que S es una matriz antisimétrica. Por consecuencia existe una propiedad que relaciona la derivada de las matrices ortonormales con las matrices antisimétricas, y puede declararse así:

$$S = \dot{R}R^{-1}. \tag{5.19}$$

Velocidad de un punto debido a una trama de referencia giratoria

Considere un vector fijo ${}^BP$ que no cambia respecto a la trama $\{B\}$. Su descripción en otra trama {A} se da así:

$${}^AP = {}^A_BR\, {}^BP. \tag{5.20}$$

Si la trama $\{B\}$ está girando (lo que implica que la derivada ${}^A_B\dot{R}$ es distinta de cero), entonces ${}^AP$ estaría cambiando aunque ${}^BP$ fuera constante; esto es,

$${}^A\dot{P} = {}^A_B\dot{R}\, {}^BP, \tag{5.21}$$

o, utilizando nuestra notación para velocidad,

$${}^AV_P = {}^A_B\dot{R}\, {}^BP. \tag{5.22}$$

Ahora vamos a rescribir la ecuación (5.22) sustituyendo para ${}^B P$, para obtener

$$ {}^A V_P = {}^A_B\dot{R}\, {}^A_B R^{-1}\, {}^A P. \tag{5.23} $$

Haciendo uso de nuestro resultado (5.19) para matrices ortonormales, tenemos que:

$$ {}^A V_P = {}^A_B S\, {}^A P, \tag{5.24} $$

en donde hemos adornado a S con subíndices y superíndices para indicar que es la matriz antisimétrica asociada con la matriz de rotación específica ${}^A_B R$. Debido a su aparición en la ecuación (5.24) y por otras razones que veremos en breve, a la matriz antisimétrica que hemos presentado se le llama **matriz de velocidad angular**.

Las matrices antisimétricas y el producto cruz de vectores

Si asignamos los elementos en una matriz antisimétrica S como

$$ S = \begin{bmatrix} 0 & -\Omega_x & \Omega_y \\ \Omega_x & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix}, \tag{5.25} $$

y definimos el siguiente vector columna de 3×1:

$$ \Omega = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}, \tag{5.26} $$

entonces puede verificarse fácilmente que

$$ SP = \Omega \times P, \tag{5.27} $$

en donde P es un vector arbitrario y $\times$ denota la operación de producto cruz o vectorial.

Al vector Ω de 3×1, que corresponde a la matriz de velocidad angular de 3×3, se le llama **vector de velocidad angular** y ya lo habíamos presentado en la sección 5.2.

Por ende, nuestra relación (5.24) puede escribirse así:

$$ {}^A V_P = {}^A\Omega_B \times {}^A P, \tag{5.28} $$

en donde hemos mostrado la notación para Ω, indicando que es el vector de velocidad angular que especifica el movimiento de la trama {B} respecto a la trama {A}.

Interpretación física del vector de velocidad angular

Habiendo concluído que existe cierto vector Ω que satisface la ecuación (5.28), podemos ahora interpretar su significado físico. Derive Ω por diferenciación directa de una matriz de rotación; esto es,

$$ \dot{R} = \lim_{\Delta t \to 0} \frac{R(t + \Delta t) - R(t)}{\Delta t}. \tag{5.29} $$

Ahora escriba $R(t + \Delta t)$ como la composición de dos matrices, a saber:

$$ R(t + \Delta t) = R_K(\Delta\theta) R(t), \tag{5.30} $$

en donde durante el intervalo Δt ocurre una pequeña rotación de $\Delta\theta$ sobre el eje $\hat{K}$. Utilizando la ecuación (5.30) escriba la (5.29) así:

$$\dot{R} = \lim_{\Delta t \to 0} \left(\frac{R_K(\Delta\theta) - I_3}{\Delta t} R(t) \right); \tag{5.31}$$

esto es,

$$\dot{R} = \left(\lim_{\Delta t \to 0} \frac{R_K(\Delta\theta) - I_3}{\Delta t} \right) R(t). \tag{5.32}$$

Ahora, basándonos en pequeñas sustituciones de ángulos en la ecuación (2.80) tenemos que

$$R_K(\Delta\theta) = \begin{bmatrix} 1 & -k_z\Delta\theta & k_y\Delta\theta \\ k_z\Delta\theta & 1 & -k_x\Delta\theta \\ -k_y\Delta\theta & k_x\Delta\theta & 1 \end{bmatrix}. \tag{5.33}$$

Así, la ecuación (5.32) puede escribirse como

$$\dot{R} = \left(\lim_{\Delta t \to 0} \frac{\begin{bmatrix} 0 & -k_z\Delta\theta & k_y\Delta\theta \\ k_z\Delta\theta & 0 & -k_x\Delta\theta \\ -k_y\Delta\theta & k_x\Delta\theta & 0 \end{bmatrix}}{\Delta t} \right) R(t). \tag{5.34}$$

Finalmente, dividiendo la matriz entre Δt y después tomando el límite, tenemos que

$$\dot{R} = \begin{bmatrix} 0 & -k_z\dot{\theta} & k_y\dot{\theta} \\ k_z\dot{\theta} & 0 & -k_x\dot{\theta} \\ -k_y\dot{\theta} & k_x\dot{\theta} & 0 \end{bmatrix} R(t). \tag{5.35}$$

Por ende, podemos ver que

$$\dot{R}R^{-1} = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix}, \tag{5.36}$$

en donde

$$\Omega = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} = \begin{bmatrix} k_x\dot{\theta} \\ k_y\dot{\theta} \\ k_z\dot{\theta} \end{bmatrix} = \dot{\theta}\hat{K}. \tag{5.37}$$

El significado físico del vector de velocidad angular Ω es que, en cualquier instante, el cambio en orientación de una trama giratoria puede verse como una rotación sobre determinado eje $\hat{K}$. Este **eje instantáneo de rotación**, si se toma como un vector unitario y después se le aplica una escala mediante la velocidad de rotación sobre ese otro eje ($\dot{\theta}$), produce el vector de velocidad angular.

Otras representaciones de la velocidad angular

Puede ser posible que existan otras representaciones de la velocidad angular; por ejemplo, imagine que la velocidad angular de un cuerpo giratorio está disponible en forma de proporciones del conjunto de ángulos de Euler Z-Y-Z:

$$\dot{\Theta}_{Z'Y'Z'} = \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}. \tag{5.38}$$

Dado este modo de representación o cualquier otro que utilice alguno de los 24 **conjuntos de ángulos**, podría ser conveniente derivar el vector de velocidad angular equivalente.

Hemos visto que

$$\dot{R}R^T = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix}. \tag{5.39}$$

A partir de esta matriz de ecuación se pueden extraer tres ecuaciones independientes:

$$\begin{aligned} \Omega_x &= \dot{r}_{31}r_{21} + \dot{r}_{32}r_{22} + \dot{r}_{33}r_{23}, \\ \Omega_y &= \dot{r}_{11}r_{31} + \dot{r}_{12}r_{32} + \dot{r}_{13}r_{33}, \\ \Omega_z &= \dot{r}_{21}r_{11} + \dot{r}_{22}r_{12} + \dot{r}_{23}r_{13}. \end{aligned} \tag{5.40}$$

De la ecuación (5.40) y una descripción simbólica de R en términos de un conjunto de ángulos, podemos obtener las expresiones que relacionan las velocidades de establecimiento de ángulos con el vector de velocidad angular equivalente. Las expresiones resultantes pueden convertirse a forma matricial; por ejemplo, para los ángulos de EulerZ-Y-Z,

$$\Omega = E_{Z'Y'Z'}(\Theta_{Z'Y'Z'})\dot{\Theta}_{Z'Y'Z'}. \tag{5.41}$$

Esto es, $E(\cdot)$ es un jacobiano que relaciona un vector de velocidad definido a partir del conjunto de ángulos con el vector de velocidad angular y es una función de los valores instantáneos del ángulo establecido. La forma de $E(\cdot)$ depende del ángulo establecido para el que haya sido desarrollado; por lo tanto, se agrega un subíndice para indicarlo.

EJEMPLO 5.2

Construya la matriz E que relacione los ángulos de Euler Z-Y-Z con el vector de velocidad angular; esto es, encuentre $E_{Z'Y'Z'}$ en la ecuación (5.41).

Utilizando las ecuaciones (2.72) y (5.40) y realizando las diferenciaciones simbólicas requeridas, tenemos que

$$E_{Z'Y'Z'} = \begin{bmatrix} 0 & -s\alpha & c\alpha s\beta \\ 0 & c\alpha & s\alpha s\beta \\ 1 & 0 & c\beta \end{bmatrix}. \tag{5.42}$$

5.5 MOVIMIENTO DE LOS VÍNCULOS DE UN ROBOT

Al considerar los movimientos de los vínculos de un robot, siempre utilizaremos la trama de vínculo {0} como nuestra trama de referencia. Por lo tanto, v_i es la velocidad lineal del origen de la trama del vínculo $\{i\}$, y ω_i es la velocidad angular de la trama del vínculo $\{i\}$.

En cualquier instante, cada uno de los vínculos de un robot en movimiento tiene cierta velocidad lineal y angular. La figura 5.6 indica estos vectores para el vínculo i. En este caso se indica que están escritos en la trama $\{i\}$.

5.6 "PROPAGACIÓN" DE VELOCIDAD DE VÍNCULO A VÍNCULO

Ahora consideremos el problema de calcular las velocidades lineal y angular de los vínculos de un robot. Un manipulador es una cadena de cuerpos, cada uno capaz de

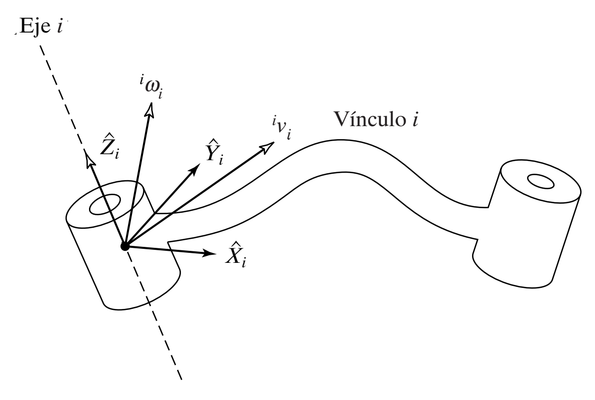


FIGURA 5.6: La velocidad del vínculo i se da mediante los vectores v_i y ω_i, que pueden escribirse en cualquier trama, incluso en $\{i\}$.

moverse en relación a sus cuerpos adyacentes. Debido a esta estructura podemos calcular la velocidad de cada vínculo en orden, empezando desde la base. La velocidad del vínculo $i + 1$ será la del vínculo i más cualquier nuevo componente de velocidad que se haya agregado por la articulación $i + 1$.[2]

Según lo que se indica en la figura 5.6, vamos a pensar ahora en cada vínculo del mecanismo como si fuera un cuerpo rígido con vectores de velocidad lineal y angular que describen su movimiento. Además, expresaremos estas velocidades respecto a la trama del vínculo en sí, en vez de hacerlo respecto al sistema de coordenadas base. La figura 5.7 muestra los vínculos i e $i + 1$, junto con sus vectores de velocidad definidos en las tramas de los vínculos.

Pueden agregarse velocidades de rotación cuando ambos vectores ω se escriben respecto a la misma trama. Por lo tanto, la velocidad angular del vínculo $i + 1$ es la misma

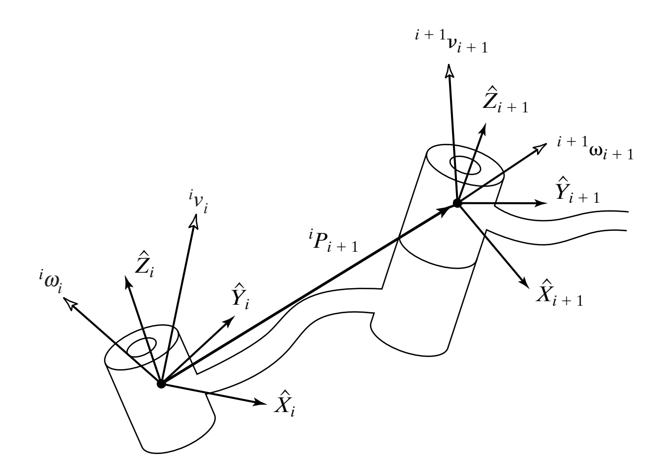


FIGURA 5.7: Vectores de velocidad de los vínculos adyacentes.

---

[2] Recuerde que la velocidad lineal se asocia con un punto, pero la velocidad angular está asociada con un cuerpo. Por lo tanto, aquí el término "velocidad de un vínculo" indica la velocidad lineal del origen de la trama del vínculo y la velocidad de rotación del mismo.

que la del vínculo i más un nuevo componente producido por la velocidad de rotación en la articulación $i + 1$. Esto puede escribirse en términos de la trama $\{i\}$ como

$$ {}^{i}\omega_{i+1} = {}^{i}\omega_i + {}^{i}_{i+1}R\,\dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}. \tag{5.43}$$

Observe que

$$ \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} = {}^{i+1}\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{bmatrix}. \tag{5.44}$$

Hemos utilizado la matriz de rotación que relaciona a las tramas $\{i\}$ e $\{i + 1\}$ para representar el componente de rotación agregado, debido al movimiento de la articulación en la trama $\{i\}$. La matriz de rotación gira el eje de rotación de la articulación $i + 1$ hacia su descripción en la trama $\{i\}$, por lo que pueden sumarse los dos componentes de velocidad angular.

Si multiplicamos previamente ambos lados de la ecuación (5.43) por ${}^{i+1}R$, podemos encontrar la descripción de la velocidad angular del vínculo $i + 1$ respecto a la trama $\{i + 1\}$:

$$ {}^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\,{}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}. \tag{5.45}$$

La velocidad lineal del origen de la trama $\{i + 1\}$ es la misma que la del origen de la trama $\{i\}$ más un nuevo componente producido por la velocidad de rotación del vínculo i. Ésta es exactamente la situación descrita por la ecuación (5.13), en donde desaparece un término debido a que ${}^{i}P_{i+1}$ es constante en la trama $\{i\}$. Por lo tanto, tenemos que

$$ {}^{i}\upsilon_{i+1} = {}^{i}\upsilon_i + {}^{i}\omega_i \times {}^{i}P_{i+1}. \tag{5.46}$$

Si multiplicamos previamente ambos lados por ${}^{i+1}_{\ i}R$, calculamos

$$ {}^{i+1}\upsilon_{i+1} = {}^{i+1}_{i}R({}^{i}\upsilon_i + {}^{i}\omega_i \times {}^{i}P_{i+1}). \tag{5.47}$$

Las ecuaciones (5.45) y (5.47) son tal vez los resultados más importantes de este capítulo. Las relaciones correspondientes para el caso en el que la articulación $i + 1$ es prismática son

$$ \begin{aligned} {}^{i+1}\omega_{i+1} &= {}^{i+1}_{i}R\,{}^{i}\omega_i, \\ {}^{i+1}\upsilon_{i+1} &= {}^{i+1}_{i}R({}^{i}\upsilon_i + {}^{i}\omega_i \times {}^{i}P_{i+1}) + \dot{d}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}. \end{aligned} \tag{5.48}$$

Si aplicamos estas ecuaciones exitosamente de vínculo en vínculo, podremos entonces calcular ${}^{N}\omega_N$ y ${}^{N}v_N$, las velocidades de rotación y lineal del último vínculo. Observe que las velocidades resultantes se expresan en términos de la trama $\{N\}$. Esto resulta útil, como veremos posteriormente. Si las velocidades se desean en términos del sistema de coordenadas base, pueden girarse hacia coordenadas base mediante la multiplicación con ${}^{0}_{N}R$.

EJEMPLO 5.3

En la figura 5.8 se muestra un manipulador de dos vínculos con articulaciones giratorias. Calcule la velocidad de la punta del brazo como función de las velocidades de articulación. Dé la respuesta en dos formas: en términos de la trama $\{3\}$ y en términos de la trama $\{0\}$.

Se ha unido la trama $\{3\}$ al extremo del manipulador, como se muestra en la figura 5.9, y deseamos encontrar la velocidad del origen de esta trama expresada en la

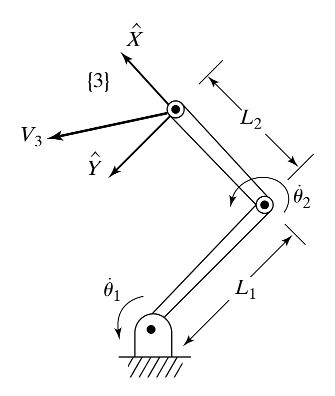


FIGURA 5.8: Un manipulador de dos vínculos.

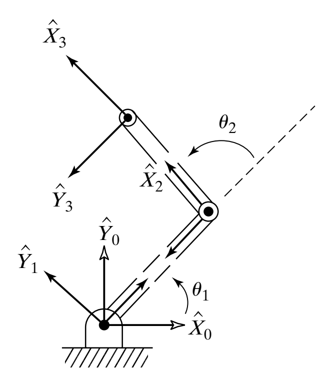


FIGURA 5.9: Asignaciones de tramas para el manipulador de dos vínculos.

trama {3}. Como segunda parte del problema expresaremos estas velocidades en la trama {0} también. Empezaremos asignando tramas a los vínculos, como hicimos antes (esto se muestra en la figura 5.9).

Utilizaremos las ecuaciones (5.45) y (5.47) para calcular la velocidad del origen de cada trama, empezando desde la trama base {0}, que tiene una velocidad de cero. Como en las ecuaciones (5.45) y (5.47) vamos a utilizar las transformaciones de vínculo, hay que calcularlas:

$$
{}^{0}_{1}T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{1}_{2}T = \begin{bmatrix} c_2 & -s_2 & 0 & l_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.49}
$$

$$
{}^{2}_{3}T = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

Observe que éstas corresponden al manipulador del ejemplo 3.3 con la articulación 3 permanentemente fija a cero grados. La transformación final entre las tramas {2} y {3} no necesita convertirse en transformación de vínculo estándar (aunque podría ser útil hacerlo). Entonces, utilizando las ecuaciones (5.45) y (5.47) en forma secuencial de vínculo en vínculo, calculamos

$$ {}^{1}\omega_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix}, \tag{5.50} $$

$$ {}^{1}\upsilon_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{5.51} $$

$$ {}^{2}\omega_2 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix}, \tag{5.52} $$

$$ {}^{2}\upsilon_2 = \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ l_1\dot{\theta}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} l_1 s_2 \dot{\theta}_1 \\ l_1 c_2 \dot{\theta}_1 \\ 0 \end{bmatrix}, \tag{5.53} $$

$$ {}^{3}\omega_3 = {}^{2}\omega_2, \tag{5.54} $$

$$ {}^{3}\upsilon_3 = \begin{bmatrix} l_1 s_2 \dot{\theta}_1 \\ l_1 c_2 \dot{\theta}_1 + l_2(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix}. \tag{5.55} $$

La ecuación (5.55) es la respuesta. Además, la velocidad de rotación de la trama {3} se encuentra en la ecuación (5.54).

Para encontrar estas velocidades respecto a la trama base inmóvil, las giramos con la matriz de rotación ${}^{0}_{3}R$:

$$ {}^{0}_{3}R = {}^{0}_{1}R \quad {}^{1}_{2}R \quad {}^{2}_{3}R = \begin{bmatrix} c_{12} & -s_{12} & 0 \\ s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.56} $$

Esta rotación produce lo siguiente:

$$ {}^{0}\upsilon_3 = \begin{bmatrix} -l_1 s_1 \dot{\theta}_1 - l_2 s_{12}(\dot{\theta}_1 + \dot{\theta}_2) \\ l_1 c_1 \dot{\theta}_1 + l_2 c_{12}(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix}. \tag{5.57} $$

Es importante señalar los dos usos para las ecuaciones (5.45) y (5.47). En primer lugar, pueden usarse como un medio para derivar expresiones analíticas, como en el ejemplo 5.3. Aquí manipulamos las ecuaciones simbólicas hasta llegar a una forma tal como (5.55), que se evaluará con una computadora en alguna aplicación. En segundo lugar, pueden usarse directamente para calcular las ecuaciones (5.45) y (5.47) a medida que se vayan escribiendo. Pueden escribirse fácilmente como una subrutina, la cual se aplica después en forma iterativa para calcular las velocidades de los vínculos. Como tales, podrían utilizarse para cualquier manipulador sin la necesidad de deducir las ecuaciones para un manipulador específico. No obstante, el cálculo produce entonces

un resultado numérico con la estructura de las ecuaciones oculta. A menudo nos interesa la estructura de un resultado analítico tal como la ecuación (5.55). Además, si nos molestamos en hacer el trabajo [esto es, las ecuaciones (5.50) a (5.57)], generalmente encontraremos que habrá menos cálculos pendientes para que los lleve a cabo la computadora en la aplicación final.

5.7 JACOBIANOS

El jacobiano es una forma multidimensional de la derivada. Por ejemplo, suponga que tenemos seis funciones, cada una de las cuales es una función de seis variables independientes:

$$\begin{aligned} y_1 &= f_1(x_1, x_2, x_3, x_4, x_5, x_6), \\ y_2 &= f_2(x_1, x_2, x_3, x_4, x_5, x_6), \\ &\vdots \\ y_6 &= f_6(x_1, x_2, x_3, x_4, x_5, x_6). \end{aligned} \tag{5.58}$$

También podríamos usar notación vectorial para escribir estas ecuaciones:

$$Y = F(X). \tag{5.59}$$

Ahora, si deseamos calcular los diferenciales de y_i en función de los diferenciales de x_j, simplemente utilizamos la regla de cadena para calcular y obtener:

$$\begin{aligned} \delta y_1 &= \frac{\partial f_1}{\partial x_1}\delta x_1 + \frac{\partial f_1}{\partial x_2}\delta x_2 + \cdots + \frac{\partial f_1}{\partial x_6}\delta x_6, \\ \delta y_2 &= \frac{\partial f_2}{\partial x_1}\delta x_1 + \frac{\partial f_2}{\partial x_2}\delta x_2 + \cdots + \frac{\partial f_2}{\partial x_6}\delta x_6, \\ &\vdots \\ \delta y_6 &= \frac{\partial f_6}{\partial x_1}\delta x_1 + \frac{\partial f_6}{\partial x_2}\delta x_2 + \cdots + \frac{\partial f_6}{\partial x_6}\delta x_6, \end{aligned} \tag{5.60}$$

lo cual podría escribirse con más simplicidad en notación vectorial:

$$\delta Y = \frac{\partial F}{\partial X}\delta X. \tag{5.61}$$

La matriz de 6×6 de derivadas parciales en la ecuación (5.61) es lo que llamamos el jacobiano, J. Observe. que si las funciones $f_1(X)$ a $f_6(X)$ son no lineales, entonces las derivadas parciales son una función de las x_i, por lo que podemos usar la notación

$$\delta Y = J(X)\delta X. \tag{5.62}$$

Al dividir ambos lados por el elemento tiempo diferencial, podemos pensar en el jacobiano como una asignación o mapeo de las velocidades en X a las velocidades en Y:

$$\dot{Y} = J(X)\dot{X}. \tag{5.63}$$

En cualquier instante específico, X tiene cierto valor y $J(X)$ es una transformación lineal. En cada nuevo instante de tiempo X ha cambiado y, por lo tanto, también ha cambiado la transformación lineal. Los jacobianos son transformaciones lineales que varían en el tiempo.

En el campo de la robótica, generalmente usamos jacobianos que relacionan velocidades de articulaciones con velocidades cartesianas en la punta del brazo; por ejemplo:

$$^{0}\nu = {}^{0}J(\Theta)\dot{\Theta}, \tag{5.64}$$

en donde Θ es el vector de ángulos de articulación del manipulador y ν es un vector de velocidades cartesianas. En la ecuación (5.64) hemos agregado un superíndice a la izquierda a nuestra notación jacobiana para indicar en qué trama se expresa la velocidad cartesiana resultante. Algunas veces el superíndice se omite cuando la trama es obvia o cuando no es importante para el desarrollo. Tenga en cuenta que, para cualquier configuración dada del manipulador, las proporciones de articulación están relacionadas con la velocidad de la punta en modo lineal, aunque ésta es solamente una relación instantánea; en el siguiente instante, el jacobiano ha cambiado ligeramente. Para el caso general de un robot con seis articulaciones el jacobiano es de 6×6, Θ es de 6×1 y $^{0}\nu$ es de 6×1. Este vector de velocidad cartesiana de 6×1 es el vector de velocidad lineal de 3×1 y el vector de velocidad de rotación de 3×1 apilados:

$$^{0}\nu = \begin{bmatrix} {}^{0}\upsilon \\ {}^{0}\omega \end{bmatrix}. \tag{5.65}$$

Pueden definirse jacobianos de cualquier dimensión (incluyendo los no cuadrados). El número de filas es igual al número de grados de libertad en el espacio cartesiano que se esté considerando. El número de columnas en un jacobiano es igual al número de articulaciones del manipulador. Por ejemplo, al tratar con un brazo planar no hay razón para que el jacobiano tenga más de tres filas, aunque para los manipuladores planares redundantes podrían haber un número arbitrariamente grande de columnas (una para cada articulación).

En el caso de un brazo de dos vínculos podemos escribir un jacobiano de 2×2 que relacione las proporciones de articulación con la velocidad del efector final. Del resultado del ejemplo 5.3 podemos fácilmente determinar el jacobiano de nuestro brazo de dos vínculos. El jacobiano escrito en la trama {3} sería entonces [de la ecuación (5.55)]:

$$^{3}J(\Theta) = \begin{bmatrix} l_1 s_2 & 0 \\ l_1 c_2 + l_2 & l_2 \end{bmatrix}, \tag{5.66}$$

y el jacobiano escrito en la trama {0} sería [de la ecuación (5.57)]:

$$^{0}J(\Theta) = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}. \tag{5.67}$$

Observe que, en ambos casos, hemos optado por escribir una matriz cuadrada que relacione las velocidades de articulación con la velocidad del efector final. También podríamos considerar un jacobiano de 3×2 que incluyera la velocidad angular del efector final.

Considerando las ecuaciones (5.58) a (5.62) que definen el jacobiano, podemos ver que éste podría encontrarse también al diferenciar directamente las ecuaciones cinemáticas del mecanismo. Esto es simple para la velocidad lineal, pero no hay un vector de orientación de 3×1 cuya derivada sea ω. Por ende, hemos introducido un método para derivar el jacobiano utilizando la aplicación sucesiva de las ecuaciones (5.45) y (5.47). Hay varios métodos más que pueden usarse (vea, por ejemplo, la referencia bibliográfica [4]), uno de los cuales se presentará posteriormente en la sección 5.8. Una razón para derivar jacobianos mediante el método presentado es que nos prepara para el material del capítulo 6, en el cual encontraremos que se aplican técnicas similares para calcular las ecuaciones dinámicas del movimiento de un manipulador.

Cómo cambiar la trama de referencia de un jacobiano

Dado un jacobiano escrito en la trama $\{B\}$, es decir,

$$\begin{bmatrix} {}^{B}\upsilon \\ {}^{B}\omega \end{bmatrix} = {}^{B}\nu = {}^{B}J(\Theta)\dot{\Theta}, \tag{5.68}$$

podría ser conveniente proporcionar una expresión para el jacobiano en otra trama, $\{A\}$. En primer lugar, observe que un vector de velocidad cartesiana de 6×1 dado en $\{B\}$ se describe en términos de $\{A\}$ mediante la siguiente transformación:

$$\begin{bmatrix} {}^{A}\upsilon \\ {}^{A}\omega \end{bmatrix} = \left[\begin{array}{c|c} {}^{A}_{B}R & 0 \\ \hline 0 & {}^{A}_{B}R \end{array}\right] \begin{bmatrix} {}^{B}\upsilon \\ {}^{B}\omega \end{bmatrix}. \tag{5.69}$$

Por ende, podemos escribir

$$\begin{bmatrix} {}^{A}\upsilon \\ {}^{A}\omega \end{bmatrix} = \left[\begin{array}{c|c} {}^{A}_{B}R & 0 \\ \hline 0 & {}^{A}_{B}R \end{array}\right] {}^{B}J(\Theta)\dot{\Theta}. \tag{5.70}$$

Ahora queda claro que para cambiar la trama de referencia de un jacobiano se utiliza la siguiente relación:

$$^{A}J(\Theta) = \left[\begin{array}{c|c} {}^{A}_{B}R & 0 \\ \hline 0 & {}^{A}_{B}R \end{array}\right] {}^{B}J(\Theta). \tag{5.71}$$

5.8 SINGULARIDADES

Si tenemos una transformación lineal que relacione la velocidad de articulación con la velocidad cartesiana, una pregunta razonable sería: ¿es invertible esta matriz? Es decir, ¿es no singular? Si la matriz es no singular, entonces podemos invertirla para calcular las velocidades de articulación a partir de las velocidades cartesianas dadas:

$$\dot{\Theta} = J^{-1}(\Theta)\nu. \tag{5.72}$$

Ésta es una importante relación. Por ejemplo, suponga que deseamos que la mano del robot se mueva con un cierto vector de velocidad en espacio cartesiano. Utilizando la ecuación (5.72) calcularíamos las proporciones de articulación necesarias en cada instante a lo largo de la ruta. La pregunta real de invertibilidad sería: ¿es invertible el jacobiano para todos los valores de Θ? Si no, ¿en dónde no es invertible?

La mayoría de los manipuladores tiene valores de Θ para los que el jacobiano se vuelve singular. Dichas ubicaciones se llaman **singularidades del mecanismo,** o simplemente **singularidades**. Todos los manipuladores tienen singularidades en los límites o frontera de su espacio de trabajo, y la mayoría tienen regiones de singularidad dentro de su espacio de trabajo. Un estudio a profundidad de la clasificación de las singularidades estaría más allá del alcance de este libro; para obtener más información, vea la referencia [5]. Para fines de este libro y sin dar definiciones rigurosas, clasificaremos las singularidades en dos categorías:

1. **Singularidades en la frontera del espacio de trabajo:** ocurren cuando el manipulador está completamente extendido o doblado sobre sí mismo, en tal forma que el efector final se encuentra en o muy cerca de los límites del espacio de trabajo.

2. **Singularidades dentro del espacio de trabajo:** ocurren lejos de los límites del espacio de trabajo; generalmente se producen por el alineamiento de dos o más ejes de articulación.

Cuando un manipulador se encuentra en una configuración singular, ha perdido uno o más grados de libertad (viéndolo desde el espacio cartesiano). Esto significa que hay cierta dirección (o subespacio) en el espacio cartesiano sobre la cual es imposible mover la mano del robot, sin importar qué proporciones de articulación estén seleccionadas. Es obvio que esto ocurre en la frontera del espacio de trabajo de los robots.

EJEMPLO 5.4

¿En dónde están las singularidades del brazo simple de dos vínculos del ejemplo 5.3? ¿Cuál es la explicación física de las singularidades? ¿Son singularidades en la frontera del espacio de trabajo o singularidades dentro del espacio de trabajo?

Para encontrar los puntos singulares de un mecanismo, debemos examinar el determinante de su jacobiano. En los puntos en los que el determinante es igual a cero, el jacobiano ha perdido todo el rango y es singular.

$$DET[J(\Theta)] = \begin{bmatrix} l_1 s_2 & 0 \\ l_1 c_2 + l_2 & l_2 \end{bmatrix} = l_1 l_2 s_2 = 0. \tag{5.73}$$

Evidentemente existe una singularidad del mecanismo cuando θ_2 es 0 o 180 grados. Físicamente, cuando $\theta_2 = 0$ el brazo está completamente extendido. En esta configuración es posible mover el efector final sobre una dirección cartesiana solamente (la que sea perpendicular al brazo). Por lo tanto, el mecanismo ha perdido un grado de libertad. De igual forma, cuando $\theta_2 = 180$ el brazo está completamente retractado y es posible moverlo de nuevo solamente en una dirección cartesiana, y no en dos. Clasificaremos a éstas como singularidades en la frontera del espacio de trabajo, ya que existen en el borde del espacio de trabajo del manipulador. Observe que el jacobiano escrito con respecto a la trama {0}, o a cualquier otra trama, hubiera producido el mismo resultado.

El peligro al aplicar la ecuación (5.72) en un sistema de control de robots es que, en un punto singular, ¡el jacobiano inverso estalla! Esto ocasiona que las velocidades de articulación se acerquen al infinito, a medida que se llega a la singularidad.

EJEMPLO 5.5

Considere el robot de dos vínculos del ejemplo 5.3, a medida que mueve su efector final a lo largo del eje $\hat{X}$ a 1.0 m/s, como se muestra en la figura 5.10. Demuestre que las proporciones de articulación son razonables cuando se está lejos de una singularidad pero que, a medida que se acerca a una singularidad cuando $\theta_2 = 0$, las proporciones de articulación tienden al infinito.

Empezaremos calculando la inversa del jacobiano escrita en {0}:

$${}^{0}J^{-1}(\Theta) = \frac{1}{l_1 l_2 s_2} \begin{bmatrix} l_2 c_{12} & l_2 s_{12} \\ -l_1 c_1 - l_2 c_{12} & -l_1 s_1 - l_2 s_{12} \end{bmatrix}. \tag{5.74}$$

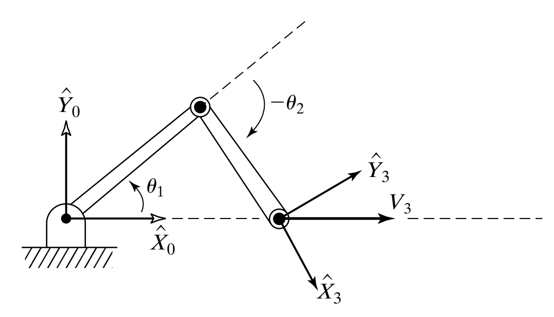


FIGURA 5.10: Un manipulador de dos vínculos que mueve su punta a una velocidad lineal constante.

Entonces, utilizando la ecuación (5.74) para una velocidad de 1 m/s en la dirección $\hat{X}$, podemos calcular las proporciones de articulación como una función de la configuración del manipulador:

$$\dot{\theta}_1 = \frac{c_{12}}{l_1 s_2}, \tag{5.75}$$

$$\dot{\theta}_2 = -\frac{c_1}{l_2 s_2} - \frac{c_{12}}{l_1 s_2}.$$

Evidentemente, a medida que el brazo se extiende hacia $\theta_2 = 0$, ambas proporciones de articulación se van a infinito.

EJEMPLO 5.6

Para el manipulador PUMA 560, proporcione dos ejemplos de singularidades que pueden ocurrir.

Hay singularidad cuando θ_3 está cerca de –90.0 grados. Dejaremos como ejercicio el cálculo del valor exacto de θ_3. (Vea el ejercicio 5.14). En esta situación, los vínculos 2 y 3 se "extienden completamente", justo igual que la ubicación singular del manipulador de dos vínculos del ejemplo 5.3. Esto se clasifica como una singularidad en los límites del espacio de trabajo.

Siempre que $\theta_5 = 0.0$ grados, el manipulador se encuentra en una configuración singular. En esta configuración, los ejes de articulación 4 y 6 se alinean; sus dos acciones producirían el mismo movimiento del efector final, por lo que es como si se hubiera perdido un grado de libertad. Como esto puede ocurrir dentro de la envoltura del espacio de trabajo, lo clasificaremos como singularidad dentro del espacio de trabajo.

5.9 FUERZAS ESTÁTICAS EN LOS MANIPULADORES

La naturaleza de tipo cadena de un manipulador nos lleva con mucha naturalidad a considerar cómo las fuerzas y los momentos se "propagan" de un vínculo al siguiente. Generalmente, el robot está empujando algo en el entorno con el extremo libre de la cadena (el efector final) o tal vez esté soportando una carga en la mano. Deseamos resolver para los momentos de torsión de las articulaciones que deben estar actuando para mantener el sistema en equilibrio estático.

Al considerar las fuerzas estáticas en un manipulador, primero bloqueamos todas las articulaciones para que el manipulador se convierta en una estructura. Después,

consideramos cada vínculo en esta estructura y escribimos una relación de balance fuerza-momento en términos de las tramas de los vínculos. Finalmente, calculamos qué momento de torsión estático debe estar actuando sobre el eje de articulación para que el manipulador pueda estar en equilibrio estático. De esta manera resolvemos para el conjunto de momentos de torsión de articulación necesarios para soportar una carga estática que actúe en el efector final.

En esta sección no consideraremos la fuerza en los vínculos debida a la gravedad (dejaremos esto para el capítulo 6). Las fuerzas estáticas y momentos de torsión que estamos considerando en las articulaciones son aquellos producidos por una fuerza estática o momento de torsión (o ambos) que actúa sobre el último vínculo; por ejemplo, como cuando el manipulador pone a su efector final en contacto con el entorno.

Definimos símbolos especiales para la fuerza y el momento de torsión ejercidos por un vínculo adyacente:

f_i = fuerza ejercida sobre el vínculo i por el vínculo $i-1$.

n_i = momento de torsión ejercido sobre el vínculo i por el vínculo $i-1$.

Utilizaremos nuestra convención usual para asignar tramas a vínculos. La figura 5.11 muestra las fuerzas estáticas y momentos (excluyendo la fuerza de gravedad) que actúan sobre el vínculo i. Si sumamos las fuerzas y las hacemos iguales a cero, tenemos

$$ {}^{i}f_i - {}^{i}f_{i+1} = 0. \tag{5.76}$$

Si sumamos los momentos de torsión sobre el origen de la trama $\{i\}$, tenemos

$$ {}^{i}n_i - {}^{i}n_{i+1} - {}^{i}P_{i+1} \times {}^{i}f_{i+1} = 0. \tag{5.77}$$

Si empezamos con una descripción de la fuerza y el momento aplicados por la mano, podemos calcular la fuerza y el momento aplicados por cada vínculo, trabajando desde el último hasta llegar a la base (vínculo 0). Para hacer esto, formulamos las expresiones de fuerza-momento (5.76) y (5.77) de tal forma que especifiquen iteraciones de los vínculos con mayor numeración hasta los vínculos con menor numeración. El resultado puede escribirse así:

$$ {}^{i}f_i = {}^{i}f_{i+1}, \tag{5.78}$$

$$ {}^{i}n_i = {}^{i}n_{i+1} + {}^{i}P_{i+1} \times {}^{i}f_{i+1}. \tag{5.79}$$

Para poder escribir estas ecuaciones sólo en términos de fuerzas y momentos definidos dentro de sus propias tramas de vínculos, transformamos con la matriz de rotación que

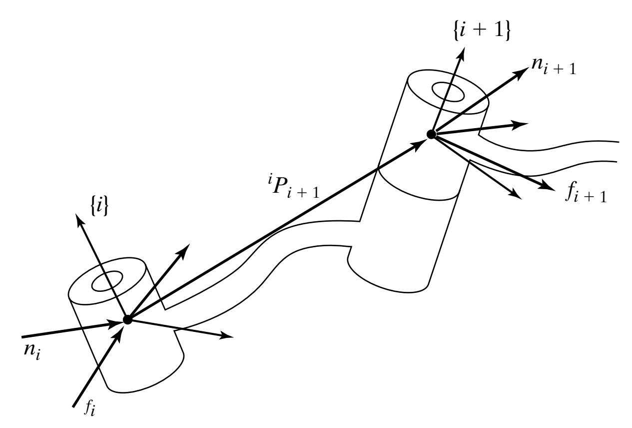


FIGURA 5.11: Balance de fuerza-momento estático para un solo vínculo.

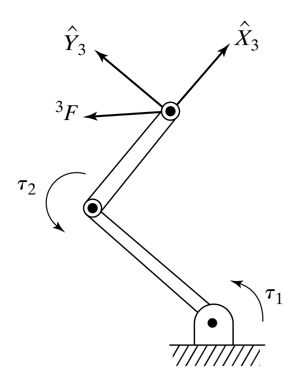


FIGURA 5.12: Un manipulador de dos vínculos, aplicando una fuerza en su punta.

describe a la trama $\{i+1\}$ respecto a la trama $\{i\}$. Esto produce nuestro resultado más importante para la "propagación" de fuerzas estáticas de vínculo en vínculo:

$$^i f_i = {}^{i}_{i+1}R \; {}^{i+1}f_{i+1}, \tag{5.80}$$

$$^i n_i = {}^{i}_{i+1}R \; {}^{i+1}n_{i+1} + {}^iP_{i+1} \times {}^i f_i. \tag{5.81}$$

Finalmente surge esta pregunta: ¿qué momentos de torsión se necesitan en las articulaciones para poder balancear las fuerzas de reacción y los momentos que actúan sobre los vínculos? Todos los componentes de los vectores de fuerza y momento son resistidos por la estructura del mecanismo en sí, excepto por el momento de torsión sobre el eje de articulación. Por lo tanto, para encontrar el momento de torsión de articulación requerido para mantener el equilibrio estático, se calcula el producto punto del vector eje de articulación con el vector de momento que actúa sobre el vínculo:

$$\tau_i = {}^i n_i^T \, {}^i\hat{Z}_i. \tag{5.82}$$

En el caso en el que la articulación i sea prismática, calculamos la fuerza del actuador de articulación así:

$$\tau_i = {}^i f_i^T \, {}^i\hat{Z}_i. \tag{5.83}$$

Observe que estamos usando el símbolo τ incluso para una fuerza de articulación lineal.

Por convención, definimos la dirección positiva del momento de torsión de una articulación como la dirección en la que se movería la articulación en la dirección del ángulo de articulación creciente.

Las ecuaciones (5.80) a la (5.83) nos proporcionan el medio para calcular los momentos de torsión de articulación necesarios para aplicar cualquier fuerza o momento con el efector final de un manipulador en el caso estático.

EJEMPLO 5.7

El manipulador de dos vínculos del ejemplo 5.3 está aplicando un vector de fuerza ^3F con su efector final. (Considere que esta fuerza actúa en el origen de $\{3\}$). Encuentre los momentos de torsión de articulación requeridos como una función de configuración y de la fuerza aplicada. (Vea la figura 5.12).

Aplicamos las ecuaciones (5.80) a (5.82), empezando desde el último vínculo y avanzando hacia la base del robot:

$$ {}^{2}f_{2} = \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix}, \tag{5.84} $$

$$ {}^{2}n_{2} = l_2 \hat{X}_2 \times \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ l_2 f_y \end{bmatrix}, \tag{5.85} $$

$$ {}^{1}f_{1} = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} = \begin{bmatrix} c_2 f_x - s_2 f_y \\ s_2 f_x + c_2 f_y \\ 0 \end{bmatrix}, \tag{5.86} $$

$$ {}^{1}n_{1} = \begin{bmatrix} 0 \\ 0 \\ l_2 f_y \end{bmatrix} + l_1 \hat{X}_1 \times {}^{1}f_1 = \begin{bmatrix} 0 \\ 0 \\ l_1 s_2 f_x + l_1 c_2 f_y + l_2 f_y \end{bmatrix}. \tag{5.87} $$

Por lo tanto, tenemos que

$$ \tau_1 = l_1 s_2 f_x + (l_2 + l_1 c_2) f_y, \tag{5.88} $$

$$ \tau_2 = l_2 f_y. \tag{5.89} $$

Esta relación puede escribirse como un operador matricial:

$$ \tau = \begin{bmatrix} l_1 s_2 & l_2 + l_1 c_2 \\ 0 & l_2 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix}. \tag{5.90} $$

¡No es coincidencia que esta matriz sea la transpuesta del jacobiano que encontramos en la ecuación (5.66)!

5.10 JACOBIANOS EN EL DOMINIO DE LA FUERZA

Hemos encontrado momentos de torsión de articulación que balancearán exactamente las fuerzas en la mano en una situación estática. Cuando actúan fuerzas sobre un mecanismo, el trabajo (en el sentido técnico) se realiza si el mecanismo se mueve a través de un desplazamiento. El trabajo se define como una fuerza que actúa a través de una distancia y es un escalar con unidades de energía. El principio de **trabajo virtual** nos permite hacer ciertas aseveraciones sobre el caso estático al permitir que la cantidad de este desplazamiento se vuelva infinitesimalmente pequeña. El trabajo tiene unidades de energía, por lo que debe ser igual si se le mide en cualquier conjunto de coordenadas generalizadas. Específicamente, podemos igualar el trabajo realizado en términos cartesianos con el trabajo realizado en términos de espacio de articulación. En el caso multidimensional, el trabajo es el producto punto de una fuerza o momento de torsión vectorial y el desplazamiento de un vector. Por lo tanto, tenemos que:

$$ \mathcal{F} \cdot \delta\chi = \tau \cdot \delta\Theta, \tag{5.91} $$

en donde $\mathcal{F}$ es un vector de fuerza-momento cartesiano de 6×1 que actúa en el efector final, $\delta\chi$ es un desplazamiento cartesiano infinitesimal de 6×1 del efector final, τ es un vector de 6×1 de momentos de torsión en las articulaciones y $\delta\Theta$ es un vector

de 6 × 1 de desplazamientos infinitesimales de articulación. La expresión (5.91) también puede escribirse así:

$$\mathcal{F}^T \delta\chi = \tau^T \delta\Theta. \tag{5.92}$$

La definición del jacobiano es

$$\delta\chi = J\delta\Theta, \tag{5.93}$$

por lo que podemos escribir

$$\mathcal{F}^T J\delta\theta = \tau^T \delta\Theta, \tag{5.94}$$

lo cual debe ser válido para todos los $\delta\Theta$; por ende, tenemos que

$$\mathcal{F}^T J = \tau^T. \tag{5.95}$$

Si trasponemos ambos lados se produce este resultado:

$$\tau = J^T \mathcal{F}. \tag{5.96}$$

La ecuación (5.96) verifica en general lo que vimos en el caso específico del manipulador de dos vínculos en el ejemplo 5.6: la transpuesta del jacobiano asigna fuerzas cartesianas que actúan en la mano, a momentos de torsión de articulación equivalentes. Cuando el jacobiano se escribe respecto a la trama {0} entonces los vectores de fuerza escritos en {0} pueden transformarse, como se clarifica mediante la siguiente notación:

$$\tau = {}^0J^T\, {}^0\mathcal{F}. \tag{5.97}$$

Cuando el jacobiano pierde el rango completo, hay ciertas direcciones en las que el efector final no puede ejercer fuerzas estáticas, aun si se desea. Esto es, en la ecuación (5.97), si el jacobiano es singular, $\mathcal{F}$ podría incrementarse o reducirse en ciertas direcciones (aquellas que definen el espacio nulo del jacobiano [6]) sin efecto en el valor calculado para τ. Esto también significa que cerca de configuraciones singulares, la ventaja mecánica tiende hacia el infinito, por lo que con pequeños momentos de torsión de articulación podrían generarse grandes fuerzas en el efector final.[3] Por lo tanto, las singularidades se manifiestan a sí mismas en el dominio de la fuerza así como en el dominio de la posición.

Observe que la ecuación (5.97) es una relación muy interesante, ya que nos permite convertir una cantidad cartesiana en una cantidad en el espacio de articulación sin calcular ninguna función cinemática inversa. Haremos uso de esto cuando consideremos el problema de control en capítulos posteriores.

5.11 TRANSFORMACIÓN CARTESIANA DE VELOCIDADES Y FUERZAS ESTÁTICAS

Podría ser conveniente pensar en términos de representaciones de 6 × 1 de la velocidad general de un cuerpo:

$$\nu = \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{5.98}$$

De igual forma podríamos considerar representaciones de 6 × 1 de vectores de fuerza generales, tales como

$$\mathcal{F} = \begin{bmatrix} F \\ N \end{bmatrix}, \tag{5.99}$$

[3] Considere un manipulador planar de dos vínculos extendido casi por completo, con el efector final en contacto con una superficie de reacción. En esta configuración, "pequeños" momentos de torsión de articulación podrían ejercer fuerzas arbitrariamente grandes.

en donde F es un vector de fuerza de 3×1 y N es un vector de momento de 3×1. Es natural entonces pensar en transformaciones de 6×6 que asignen estas cantidades de una trama a otra. Esto es exactamente lo que ya hemos hecho al considerar la propagación de velocidades y fuerzas de un vínculo a otro. Aquí escribimos las ecuaciones (5.45) y (5.47) en forma de operador matricial para transformar vectores de velocidad generales en la trama {A} a su descripción en la trama {B}.

Las dos tramas involucradas aquí están conectadas rígidamente, por lo que $\dot{\theta}_{i+1}$, que aparece en la ecuación (5.45), se hace cero al derivar la relación

$$\begin{bmatrix} {}^{B}\upsilon_B \\ {}^{B}\omega_B \end{bmatrix} = \begin{bmatrix} {}^{B}_{A}R & -{}^{B}_{A}R\,{}^{A}P_{BORG}\times \\ 0 & {}^{B}_{A}R \end{bmatrix} \begin{bmatrix} {}^{A}\upsilon_A \\ {}^{A}\omega_A \end{bmatrix}, \tag{5.100}$$

en donde se entiende que el producto cruz es el operador matricial

$$P\times = \begin{bmatrix} 0 & -p_x & p_y \\ p_x & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix}. \tag{5.101}$$

Ahora bien, la ecuación (5.100) relaciona las velocidades en una trama con las de otra, por lo que al operador de 6×6 lo llamaremos una **transformación de velocidad**; utilizaremos el símbolo T_v. En este caso es una transformación de velocidad que asigna velocidades en {A} a velocidades en {B}, por lo que usaremos la siguiente notación para expresar la ecuación (5.100) en forma compacta:

$${}^{B}\nu_B = {}^{B}_{A}T_\upsilon\,{}^{A}\nu_A. \tag{5.102}$$

Podemos invertir la ecuación (5.100) para poder calcular la descripción de la velocidad en términos de {A}, dadas las cantidades en {B}:

$$\begin{bmatrix} {}^{A}\upsilon_A \\ {}^{A}\omega_A \end{bmatrix} = \begin{bmatrix} {}^{A}_{B}R & {}^{A}P_{BORG}\times {}^{A}_{B}R \\ 0 & {}^{A}_{B}R \end{bmatrix} \begin{bmatrix} {}^{B}\upsilon_B \\ {}^{B}\omega_B \end{bmatrix}, \tag{5.103}$$

o

$${}^{A}\nu_A = {}^{A}_{B}T_\upsilon\,{}^{B}\nu_B. \tag{5.104}$$

Observe que estas asignaciones de velocidades de trama a trama dependen de ${}^{A}_{B}T$ (o de su inversa) y, por lo tanto, deben interpretarse como resultados instantáneos, a menos que la relación entre las dos tramas sea estática. De manera similar, de las ecuaciones (5.80) y (5.81), escribimos la matriz de 6×6 que transforma vectores de fuerza generales en términos de {B} en su descripción en la trama {A}:

$$\begin{bmatrix} {}^{A}F_A \\ {}^{A}N_A \end{bmatrix} = \begin{bmatrix} {}^{A}_{B}R & 0 \\ {}^{A}P_{BORG}\times {}^{A}_{B}R & {}^{A}_{B}R \end{bmatrix} \begin{bmatrix} {}^{B}F_B \\ {}^{B}N_B \end{bmatrix}, \tag{5.105}$$

que puede escribirse en forma compacta así:

$${}^{A}\mathcal{F}_A = {}^{A}_{B}T_f\,{}^{B}\mathcal{F}_B, \tag{5.106}$$

en donde T_f denota una **transformación de fuerza-momento**.

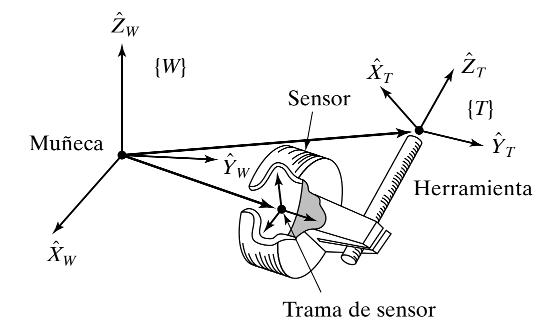


FIGURA 5.13: Tramas de interés con un sensor de fuerza.

Las transformaciones de velocidad y de fuerza son similares a los jacobianos en cuanto a que relacionan velocidades y fuerzas en distintos sistemas de coordenadas. Al igual que con los jacobianos, tenemos que

$$ {}_B^A T_f = {}_B^A T_\upsilon^T , \tag{5.107}$$

como puede verificarse si se examinan las ecuaciones (5.105) y (5.103).

EJEMPLO 5.8

La figura 5.13 muestra un efector final sosteniendo una herramienta. En el punto en el que el efector final se une al manipulador está ubicada una muñeca con sensor de fuerza. Éste es un dispositivo que puede medir las fuerzas y los momentos de torsión que se le aplican.

Considere que la salida de este sensor es un vector de 6 × 1 llamado ${}^S\mathcal{F}$, compuesto de tres fuerzas y tres momentos de torsión expresados en la trama del sensor, $\{S\}$. Nuestro verdadero interés está en conocer las fuerzas y momentos de torsión que se aplican en la punta de la herramienta, ${}^T\mathcal{F}$. Encuentre la transformación de 6 × 6 que transforma el vector de fuerza-momento de $\{S\}$ a la trama herramienta, $\{T\}$. La transformada que relaciona $\{T\}$ con $\{S\}$, ${}_T^S T$, es conocida. (Observe que $\{S\}$ aquí es la trama del sensor, no la trama estación).

Ésta es simplemente una aplicación de la ecuación (5.106). En primer lugar, de ${}_T^S T$ calculamos la inversa, ${}_S^T T$, que está compuesta de ${}_S^T R$ y de ${}^T P_{SORG}$. Entonces aplicamos la ecuación (5.106) para obtener

$$ {}^T\mathcal{F}_T = {}_S^T T_f \; {}^S\mathcal{F}_S , \tag{5.108}$$

en donde

$$ {}_S^T T_f = \begin{bmatrix} {}_S^T R & 0 \\ {}^T P_{SORG} \times {}_S^T R & {}_S^T R \end{bmatrix} . \tag{5.109}$$

BIBLIOGRAFÍA

[1] K. Hunt, *Kinematic Geometry of Mechanisms*, Oxford University Press, Nueva York, 1978.

[2] K. R. Symon, *Mechanics*, 3a. Edición, Addison-Wesley, Reading, MA, 1971.

[3] I. Shames, *Engineering Mechanics*, 2a. Edición, Prentice-Hall, Englewood Cliffs, NJ, 1967.

[4] D. Orin y W. Schrader, "Efficient Jacobian Determination for Robot Manipulators", en *Robotics Research: The First International Symposium*, M. Brady y R.P. Paul, Editores, MIT Press, Cambridge, MA, 1984.

[5] B. Gorla y M. Renaud, *Robots Manipulateurs*, Cepadues-Editions, Toulouse, 1984.

[6] B. Noble, *Applied Linear Algebra*, Prentice-Hall, Englewood Cliffs, NJ, 1969.

[7] J. K. Salisbury y J. Craig, "Articulated Hands: Kinematic and Force Control Issues", *International Journal of Robotics Research*, Vol. 1, Núm. 1, Primavera de 1982.

[8] C. Wampler, "Wrist Singularities: Theory and Practice", en *The Robotics Review* 2, O. Khatib, J. Craig y T. Lozano-Perez, Editores, MIT Press, Cambridge, MA, 1992.

[9] D.E. Whitney, "Resolved Motion Rate Control of Manipulators and Human Prostheses", *IEEE Transactions on Man-Machine Systems,* 1969.

EJERCICIOS

5.1 [10] Repita el ejemplo 5.3, pero utilice el jacobiano escrito en la trama {0}. ¿Los resultados son los mismos que los del ejemplo 5.3?

5.2 [25] Encuentre el jacobiano del manipulador con tres grados de libertad del ejercicio 3 del capítulo 3. Escríbalo en términos de una trama {4} ubicada en la punta de la mano y con la misma orientación que la trama {3}.

5.3 [35] Encuentre el jacobiano del manipulador con tres grados de libertad del ejercicio 3 del capítulo 3. Escríbalo en términos de la trama {4} ubicada en la punta de la mano y con la misma orientación que la trama {3}. Derive el jacobiano de tres maneras distintas: propagación de velocidad de la base a la punta, propagación de fuerza estática de la punta a la base y diferenciación directa de las ecuaciones cinemáticas.

5.4 [8] Demuestre que las singularidades en el dominio de la fuerza existen en las mismas configuraciones que las singularidades en el dominio de la posición.

5.5 [39] Calcule el jacobiano del PUMA 560 en la trama {6}.

5.6 [47] ¿Es cierto que cualquier mecanismo con tres articulaciones angulares y longitudes de vínculo distintas de cero debe tener una región con puntos singulares dentro de su espacio de trabajo?

5.7 [7] Bosqueje la figura de un mecanismo con tres grados de libertad, cuyo jacobiano de velocidad lineal sea la matriz de identidad de 3×3 sobre todas las configuraciones del manipulador. Describa la cinemática en una oración o dos.

5.8 [18] Algunas veces los mecanismos generales tienen ciertas configuraciones, llamadas "puntos isotrópicos", en donde las columnas del jacobiano se vuelven ortogonales y de igual magnitud [7]. Para el manipulador de dos vínculos del ejemplo 5.3, encuentre si existen puntos isotrópicos. Sugerencia: ¿hay un requerimiento sobre l_1 y l_2?

5.9 [50] Encuentre las condiciones necesarias para que existan puntos isotrópicos en un manipulador general con seis grados de libertad. (Vea el ejercicio 5.8).

5.10 [7] Para el manipulador de dos vínculos del ejemplo 5.2, proporcione la transformación que asigne los momentos de torsión de articulación a un vector de fuerza de 2×1, ${}^3F$, en la mano.

5.11 [14] Dada

$$
{}^A_BT = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 10.0 \\ 0.500 & 0.866 & 0.000 & 0.0 \\ 0.000 & 0.000 & 1.000 & 5.0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

si el vector de velocidad en el origen de $\{A\}$ es

$$^{A}\nu = \begin{bmatrix} 0.0 \\ 2.0 \\ -3.0 \\ 1.414 \\ 1.414 \\ 0.0 \end{bmatrix},$$

Encuentre el vector de velocidad de 6×1 cuyo punto de referencia sea el origen de $\{B\}$.

5.12 [15] Para el manipulador de tres vínculos del ejercicio 3.3, proporcione un conjunto de ángulos de articulación para los que el manipulador esté en una singularidad en la frontera del espacio de trabajo, y otro conjunto de ángulos para los cuales el manipulador esté en una singularidad dentro del espacio de trabajo.

5.13 [9] Cierto manipulador de dos vínculos tiene el siguiente jacobiano:

$$^{0}J(\Theta) = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}.$$

Ignorando la gravedad, ¿cuáles son los momentos de torsión de articulación requeridos para que el manipulador aplique un vector de fuerza estático $^{0}F = 10\hat{X}_0$?

5.14 [18] Si el parámetro de vínculo a_3 del PUMA 560 fuera cero, ocurriría una singularidad en la frontera del espacio de trabajo cuando $\theta_3 = -90.0°$. Dé una expresión para el valor de θ_3 en donde ocurra la singularidad y demuestre que, si a_3 fuera cero, el resultado sería $\theta_3 = -90.0°$. *Sugerencia*: en esta configuración pasa una línea recta a través de los ejes de articulación 2 y 3, y el punto en el que se intersecan los ejes 4, 5 y 6.

5.15 [24] Proporcione el jacobiano de 3×3 que calcule la velocidad lineal de la punta de la herramienta a partir de las tres proporciones de articulación para el manipulador del ejemplo 3.4 en el capítulo 3. Proporcione el jacobiano en la trama $\{0\}$.

5.16 [20] Un manipulador $3R$ tiene una cinemática que corresponde exactamente al conjunto de ángulos de Euler Z-Y-Z (es decir, la cinemática directa se proporciona mediante la ecuación (2.72) con $\alpha = \theta_1$, $\beta = \theta_2$ y $\gamma = \theta_3$). Proporcione el jacobiano que relaciona las velocidades de articulación con la velocidad angular del último vínculo.

5.17 [31] Imagine que para un robot de 6 grados de libertad (6-DOF) general tenemos disponibles $^{0}\hat{Z}_i$ y $^{0}P_{iorg}$ para todas las i; esto es, conocemos los valores para los vectores Z unitarios de cada trama de vínculo en términos de la trama base y conocemos las ubicaciones de los orígenes de todas las tramas de los vínculos en términos de la trama base. Digamos también que estamos interesados en la velocidad del punto de la herramienta (fijo respecto al vínculo n) y que conocemos también $^{0}P_{herramienta}$. Ahora, para una articulación angular, la velocidad de la punta de la herramienta debida a la velocidad de la articulación i se da mediante

$$^{0}\upsilon_i = \dot{\theta}_i \, {}^{0}\hat{Z}_i \times ({}^{0}P_{herramienta} - {}^{0}P_{iorg}) \tag{5.110}$$

y la velocidad angular del vínculo n debida a la velocidad de esta articulación se da mediante

$$^{0}\omega_i = \dot{\theta}_i \, {}^{0}\hat{Z}_i. \tag{5.111}$$

La velocidad lineal y angular total de la herramienta se da mediante la suma de $^{0}v_i$ y $^{0}\omega_i$ respectivamente. Proporcione ecuaciones análogas a (5.110) y (5.111) para el caso de la articulación i prismática, y escriba la matriz jacobiana de 6×6 de un manipulador 6-DOF arbitrario en términos de $\hat{Z}_i$, P_{iorg} y $^{0}P_{herramienta}$.

5.18 [18] El comportamiento cinemático de un robot $3R$ se da mediante

$$^{0}_{3}T = \begin{bmatrix} c_1 c_{23} & -c_1 s_{23} & s_1 & l_1 c_1 + l_2 c_1 c_2 \\ s_1 c_{23} & -s_1 s_{23} & -c_1 & l_1 s_1 + l_2 s_1 c_2 \\ s_{23} & c_{23} & 0 & l_2 s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Encuentre ${}^{0}J(\Theta)$ que, al multiplicarse por el vector de velocidad de articulación, da la velocidad lineal del origen de la trama {3} en forma relativa a la trama {0}.

5-19 [15] La posición del origen del vínculo 2 para un manipulador *RP* se da mediante

$$ {}^{0}P_{2ORG} = \begin{bmatrix} a_1c_1 - d_2s_1 \\ a_1s_1 + d_2c_1 \\ 0 \end{bmatrix}. $$

Proporcione el jacobiano de 2×2 que relacione las dos velocidades de articulación con la velocidad lineal del origen de la trama {2}. Proporcione un valor de Θ en el que el dispositivo esté en una singularidad.

5.20 [20] Explique lo que podría significar el enunciado: "Un manipulador de n grados de libertad (n-DOF) en una singularidad puede tratarse como un manipulador redundante en un espacio con una dimensión de $n - 1$".

EJERCICIO DE PROGRAMACIÓN (PARTE 5)

1. Dos tramas {A} y {B} no se están moviendo en forma relativa entre sí; esto es, ${}^{A}_{B}T$ es constante. En el caso planar definimos la velocidad de la trama {A} como

$$ {}^{A}v_A = \begin{bmatrix} {}^{A}\dot{x}_A \\ {}^{A}\dot{y}_A \\ {}^{A}\dot{\theta}_A \end{bmatrix}. $$

Escriba una rutina que, dadas ${}^{A}_{B}T$ y ${}^{A}v_A$, calcule ${}^{B}v_B$. *Sugerencia*: ésta es la análoga planar de (5.100). Use un encabezado de procedimiento como el siguiente (o su equivalente en C):

```
Procedure Veltrans (VAR brela: trama; VAR vrela, vrelb: vec3);
```

en donde "vrela" es la velocidad relativa a la trama {A}, o ${}^{A}v_A$, y "vrelb" es la salida de la rutina (la velocidad relativa a la trama {B}), o ${}^{B}v_B$.

2. Determine el jacobiano de 3×3 del manipulador planar de tres vínculos (del ejemplo 3.3). Para poder derivar el jacobiano deberá utilizar un análisis de propagación de velocidad (como en el ejemplo 5.2) o un análisis de fuerza estática (como en el ejemplo 5.6). Entregue su trabajo mostrando cómo obtuvo el jacobiano.

Escriba una rutina para calcular el jacobiano en la trama {3}; esto es, ${}^{3}J(\Theta)$, como una función de los ángulos de articulación. Observe que la trama {3} es la trama de vínculo estándar con origen en el eje de la articulación 3. Use un encabezado de procedimiento similar al siguiente (o su equivalente en C):

```
Procedure jacobiano (VAR theta: vec3; Var Jac: mat33);
```

Los datos del manipulador son $l_2 = l_2 = 0.5$ metros.

3. Una trama herramienta y una trama estación son definidas de la siguiente manera por el usuario para cierta tarea (las unidades están en metros y grados):

$$ {}^{W}_{T}T = [x\ y\ \theta] = [0.1\ 0.2\ 30.0], $$

$$ {}^{B}_{S}T = [x\ y\ \theta] = [0.0\ 0.0\ 0.0]. $$

En cierto instante, la punta de la herramienta se encuentra en la posición

$$ {}^{S}_{T}T = [x\ y\ \theta] = [0.6\ \ -0.3\ 45.0]. $$

En el mismo instante se miden las proporciones de articulación (en grados/segundo):

$$\dot{\Theta} = [\dot{\theta}_1\ \dot{\theta}_2\ \dot{\theta}_3] = [20.0\ -10.0\ 12.0].$$

Calcule la velocidad lineal y angular de la punta de la herramienta referida a su propia trama, ${}^{T}v_T$. Si hay más que una posible respuesta, calcúlelas todas.

EJERCICIO MATLAB 5

Este ejercicio se enfoca en la matriz jacobiana y la determinante, el control simulado con velocidad resuelta y la estática inversa para el robot planar 3-DOF, 3R. (Vea las figuras 3.6 y 3.7; los parámetros de DH se proporcionan en la figura 3.8).

El método de control con velocidad resuelta [9] se basa en la ecuación de velocidad del manipulador ${}^{k}\dot{X} = {}^{k}J\dot{\Theta}$, en donde ${}^{k}J$ es la matriz jacobiana, $\dot{\Theta}$ es el vector de velocidades relativas de articulación, ${}^{k}\dot{X}$es el vector de velocidades cartesianas comandadas (tanto de traslación como de rotación), y k es la trama de expresión para la matriz jacobiana y las velocidades cartesianas. Esta figura muestra un diagrama de bloques para simular el algoritmo de control con velocidad resuelta:

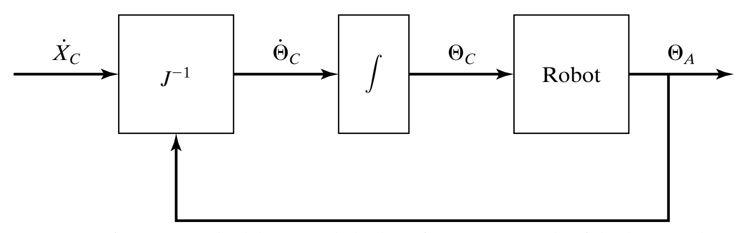


Diagrama de bloque del algoritmo con velocidad resuelta

Como se puede ver en la figura, el algoritmo de velocidad resuelta calcula las velocidades de articulación comandadas $\dot{\Theta}_C$ que se requieren para proporcionar las velocidades cartesianas comandadas $\dot{X}_C$; este diagrama debe calcularse en cada intervalo simulado. La matriz jacobiana cambia con la configuración Θ_A. Para fines de simulación, suponga que los ángulos de articulación comandados Θ_C son siempre idénticos a los ángulos de articulación realmente alcanzados, Θ_A (un resultado que raras veces es verdadero en el mundo real). Para el robot planar 3-DOF, 3R asignado, las ecuaciones de velocidad ${}^{k}\dot{X} = {}^{k}J\dot{\Theta}$ para $k = 0$ son

$${}^{0}\begin{Bmatrix} \dot{x} \\ \dot{y} \\ \omega_z \end{Bmatrix} = {}^{0}\begin{bmatrix} -L_1s_1 - L_2s_{12} - L_3s_{123} & -L_2s_{12} - L_3s_{123} & -L_3s_{123} \\ L_1c_1 + L_2c_{12} + L_3c_{123} & L_2c_{12} + L_3c_{123} & L_3c_{123} \\ 1 & 1 & 1 \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix},$$

en donde $s_{123} = \mathrm{sen}(\theta_1 + \theta_2 + \theta_3)$, $c_{123} = \cos(\theta_1 + \theta_2 + \theta_3)$, y así sucesivamente. Observe que ${}^{0}\dot{X}$ proporciona las velocidades cartesianas del origen de la trama de la mano (en el centro de las pinzas en la figura 3.6) respecto al origen de la trama base {0}, expresada en coordenadas de {0}.

Ahora bien, la mayoría de los robots industriales no pueden comandar $\dot{\Theta}_C$ directamente, por lo que primero debemos integrar estas velocidades relativas de articulación comandadas con los ángulos de articulación comandados Θ_C, que pueden ordenarse al robot en cada intervalo de tiempo. En la práctica, el esquema de integración más simple posible funciona bien, suponiendo un intervalo de control Δt pequeño: $\Theta_{\text{nuevo}} = \Theta_{\text{anterior}} + \dot{\Theta}\,\Delta t$. En su simulación de velocidad resuelta de MATLAB suponga que el robot virtual puede lograr perfectamente el Θ_{nuevo} comandado. (En los capítulos 6 y 9 presentaremos material de dinámica y control para el que no tendremos que hacer esta suposición simplificadora.) Ase-

gúrese de actualizar la matriz jacobiana con la nueva configuración Θ_{nuevo} antes de completar los cálculos de velocidad resuelta para el siguiente intervalo de tiempo.

Desarrolle un programa en MATLAB para calcular la matriz jacobiana y para simular el control con velocidad resuelta para el robot 3R planar. Dadas las longitudes del robot $L_1 = 4$, $L_2 = 3$ y $L_3 = 2$(m); los ángulos de articulación iniciales $\Theta = \{\theta_1 \ \ \theta_2 \ \ \theta_3\}^T = \{10° \ 20° \ 30°\}^T$ y las proporciones cartesianas comandadas constantes ${}^0\{\dot{X}\} = \{\dot{x} \ \ \dot{y} \ \ \omega_z\}^T = \{0.2 \ -0.3 \ -0.2\}^T$ (m/s, m/s, rad/s), simule para 5 segundos exactamente, usando intervalos de exactamente $dt = 0.1$ seg. En el mismo ciclo del programa calcule el problema de estática inversa; es decir, los momentos de torsión de articulación $T = \{\tau_1 \ \tau_2 \ \tau_3\}^T$ (Nm), dada la llave de torsión cartesiana constante comandada ${}^0\{W\} = \{f_x \ \ f_y \ \ m_z\}^T = \{1 \ 2 \ 3\}^T$ (N, N, Nm). Además, en el mismo ciclo, anime el robot en la pantalla durante cada intervalo de tiempo, de manera que pueda observar el movimiento simulado para verificar que sea correcto.

a) Para los números específicos asignados, presente cinco diagramas (cada uno en un gráfico separado, por favor):

1. Las tres proporciones de articulación activas $\dot{\Theta} = \{\dot{\theta}_1 \ \ \dot{\theta}_2 \ \ \dot{\theta}_3\}^T$ contra el tiempo.
2. Los tres ángulos de articulación activos $\Theta = \{\theta_1 \ \ \theta_2 \ \ \theta_3\}^T$ contra el tiempo.
3. Los tres componentes cartesianos de ${}^0_H T$, $X = \{x \ y \ \phi\}^T$ (rad está bien para ϕ, de manera que pueda ponerse en el mismo diagrama) contra el tiempo.
4. El determinante de la matriz jacobiana $|J|$ contra el tiempo; comente la cercanía de singularidades durante el movimiento simulado con velocidad resuelta.
5. Los tres momentos de torsión activos $T = \{\tau_1 \ \tau_2 \ \tau_3\}^T$ contra el tiempo.

Etiquete cuidadosamente (a mano está bien) cada componente en cada trazo; también etiquete los ejes con nombres y unidades.

b) Compruebe los resultados de su matriz jacobiana para los conjuntos inicial y final de ángulos de articulación mediante la Caja de Herramientas de Robótica MATLAB de Corke. Pruebe la función *jacob0()*. **Precaución:** Las funciones jacobianas de la caja de herramientas son para el movimiento de {3} respecto a {0}, no para {H} respecto a {0}, como en la asignación del problema. La anterior función da el resultado jacobiano en coordenadas de {0}; *jacobn()* da los resultados en coordenadas de {3}.

C A P Í T U L O 6

Dinámica de manipuladores

6.1 INTRODUCCIÓN

Hasta ahora, nuestro estudio de los manipuladores se ha enfocado solamente en las consideraciones cinemáticas. Hemos estudiado posiciones estáticas, fuerzas estáticas y velocidades; pero no hemos considerado *las fuerzas requeridas para producir movimiento*. En este capítulo consideraremos las ecuaciones de movimiento para un manipulador: la manera en que se produce el movimiento del manipulador debido a los momentos de torsión aplicados por los actuadores, o a fuerzas externas aplicadas al manipulador.

La dinámica de mecanismos es un campo sobre el cual se han escrito muchos libros y podríamos pasar años estudiando ese campo. Obviamente no podemos cubrir el material con todo el detalle que merece; no obstante, ciertas formulaciones del problema de la dinámica parecen especialmente adecuadas para aplicarse a los manipuladores; en especial, los métodos que hacen uso de la naturaleza de cadena serial de los manipuladores son candidatos naturales para nuestro estudio.

Hay dos problemas que deseamos resolver que están relacionados con la dinámica de un manipulador. En el primer problema se nos proporciona un punto de trayectoria, Θ, $\dot{\Theta}$ y $\ddot{\Theta}$, y deseamos encontrar el vector requerido de momentos de torsión de articulación, τ. Esta formulación de la dinámica es útil para el problema de controlar el manipulador (capítulo 10). El segundo problema es calcular cómo se moverá el mecanismo bajo la aplicación de un conjunto de momentos de torsión de articulación. Esto es, dado un vector de momento de torsión τ, calcular el movimiento resultante del manipulador, Θ, $\dot{\Theta}$ y $\ddot{\Theta}$. Esto es útil para simular el manipulador.

6.2 ACELERACIÓN DE UN CUERPO RÍGIDO

Ahora extenderemos nuestro análisis del movimiento de cuerpos rígidos al caso de las aceleraciones. En cualquier instante, los vectores de velocidad lineal y angular tienen derivadas llamadas aceleraciones lineal y angular, respectivamente. Esto es,

$$^{B}\dot{V}_{Q} = \frac{d}{dt}\,^{B}V_{Q} = \lim_{\Delta t \to o} \frac{^{B}V_{Q}(t+\Delta t) - {}^{B}V_{Q}(t)}{\Delta t}, \tag{6.1}$$

y

$$^{A}\dot{\Omega}_{B} = \frac{d}{dt}\,^{A}\Omega_{B} = \lim_{\Delta t \to o} \frac{^{A}\Omega_{B}(t+\Delta t) - {}^{A}\Omega_{B}(t)}{\Delta t}. \tag{6.2}$$

Al igual que con las velocidades, cuando se sabe que la trama de referencia de la diferenciación es cierta trama de referencia universal, por ejemplo $\{U\}$, utilizamos la notación

$$\dot{v}_{A} = {}^{U}\dot{V}_{AORG} \tag{6.3}$$

y

$$\dot{\omega}_{A} = {}^{U}\dot{\Omega}_{A}. \tag{6.4}$$

Aceleración lineal

Empezaremos reformulando la ecuación (5.12) un resultado importante del capítulo 5, que describe la velocidad de un vector ^{B}Q visto desde la trama $\{A\}$ cuando los orígenes son coincidentes:

$$^{A}V_{Q} = {}^{A}_{B}R\,^{B}V_{Q} + {}^{A}\Omega_{B} \times {}^{A}_{B}R\,^{B}Q. \tag{6.5}$$

El lado izquierdo de esta ecuación describe cómo cambia ^{A}Q en el tiempo. Debido a que los orígenes son coincidentes, podríamos reescribir la ecuación (6.5) como

$$\frac{d}{dt}({}^{A}_{B}R\,^{B}Q) = {}^{A}_{B}R\,^{B}V_{Q} + {}^{A}\Omega_{B} \times {}^{A}_{B}R\,^{B}Q. \tag{6.6}$$

Esta forma de la ecuación será útil cuando se derive la ecuación de aceleración correspondiente.

Al diferenciar la ecuación (6.5) podemos derivar expresiones para la aceleración de ^{B}Q visto desde $\{A\}$ cuando coinciden los orígenes de $\{A\}$ y $\{B\}$:

$$^{A}\dot{V}_{Q} = \frac{d}{dt}({}^{A}_{B}R\,^{B}V_{Q}) + {}^{A}\dot{\Omega}_{B} \times {}^{A}_{B}R\,^{B}Q + {}^{A}\Omega_{B} \times \frac{d}{dt}({}^{A}_{B}R\,^{B}Q). \tag{6.7}$$

Ahora aplicamos la ecuación (6.6) dos veces; una al primer término y otra al último término. El lado derecho de la ecuación (6.7) se convierte en:

$$\begin{aligned} {}^{A}_{B}R\,^{B}\dot{V}_{Q} \;&+\; {}^{A}\Omega_{B} \times {}^{A}_{B}R\,^{B}V_{Q} + {}^{A}\dot{\Omega}_{B} \times {}^{A}_{B}R\,^{B}Q \\ &+\; {}^{A}\Omega_{B} \times ({}^{A}_{B}R\,^{B}V_{Q} + {}^{A}\Omega_{B} \times {}^{A}_{B}R\,^{B}Q). \end{aligned} \tag{6.8}$$

Si combinamos dos términos obtenemos

$${}^{A}_{B}R\,^{B}\dot{V}_{Q} + 2\,{}^{A}\Omega_{B} \times {}^{A}_{B}R\,^{B}V_{Q} + {}^{A}\dot{\Omega}_{B} \times {}^{A}_{B}R\,^{B}Q + {}^{A}\Omega_{B} \times ({}^{A}\Omega_{B} \times {}^{A}_{B}R\,^{B}Q). \tag{6.9}$$

Por último, para generalizar en el caso en que los orígenes no sean coincidentes, agregamos un término que proporciona la aceleración lineal del origen de $\{B\}$, lo que produce la siguiente fórmula general para el lado derecho de la ecuación:

$$
\begin{aligned}
{}^{A}\dot{V}_{BORG} + {}^{A}_{B}R\, {}^{B}\dot{V}_{Q} + 2\,{}^{A}\Omega_{B} \times {}^{A}_{B}R\, {}^{B}V_{Q} + {}^{A}\dot{\Omega}_{B} \times {}^{A}_{B}R\, {}^{B}Q \\
+ {}^{A}\Omega_{B} \times ({}^{A}\Omega_{B} \times {}^{A}_{B}R\, {}^{B}Q).
\end{aligned}
\tag{6.10}
$$

Un caso particular que vale la pena mencionar es cuando ${}^{B}Q$ es constante, o

$$
{}^{B}V_{Q} = {}^{B}\dot{V}_{Q} = 0. \tag{6.11}
$$

En este caso, la ecuación (6.10) se simplifica a

$$
{}^{A}\dot{V}_{Q} = {}^{A}\dot{V}_{BORG} + {}^{A}\Omega_{B} \times ({}^{A}\Omega_{B} \times {}^{A}_{B}R\, {}^{B}Q) + {}^{A}\dot{\Omega}_{B} \times {}^{A}_{B}R\, {}^{B}Q. \tag{6.12}
$$

Utilizaremos este resultado para calcular la aceleración lineal de los vínculos de un manipulador con articulaciones giratorias. Cuando exista una articulación prismática se utilizará la forma más general de la ecuación (6.10).

Aceleración angular

Considere el caso en el que $\{B\}$ está girando de manera relativa a $\{A\}$ con ${}^{A}\Omega_{B}$, y $\{C\}$ está girando de manera relativa a $\{B\}$ con ${}^{B}\Omega_{C}$. Para calcular ${}^{A}\Omega_{C}$ sumamos los vectores en la trama $\{A\}$:

$$
{}^{A}\Omega_{C} = {}^{A}\Omega_{B} + {}^{A}_{B}R\, {}^{B}\Omega_{C}. \tag{6.13}
$$

Al diferenciar obtenemos

$$
{}^{A}\dot{\Omega}_{C} = {}^{A}\dot{\Omega}_{B} + \frac{d}{dt}({}^{A}_{B}R\, {}^{B}\Omega_{C}). \tag{6.14}
$$

Ahora, si aplicamos la ecuación (6.6) al último término de la ecuación (6.14), obtenemos

$$
{}^{A}\dot{\Omega}_{C} = {}^{A}\dot{\Omega}_{B} + {}^{A}_{B}R\, {}^{B}\dot{\Omega}_{C} + {}^{A}\Omega_{B} \times {}^{A}_{B}R\, {}^{B}\Omega_{C}. \tag{6.15}
$$

Utilizaremos este resultado para calcular la aceleración angular de los vínculos de un manipulador.

6.3 DISTRIBUCIÓN DE LA MASA

A menudo hablamos de la masa de un cuerpo rígido en sistemas con un solo grado de libertad; en el caso del movimiento giratorio sobre un solo eje es muy común el término *momento de inercia*. Para un cuerpo rígido que tiene la libertad de moverse en tres dimensiones, hay un número infinito de posibles ejes de rotación. En el caso de la rotación sobre un eje arbitrario necesitamos una forma completa de caracterizar la distribución de la masa de un cuerpo rígido. Aquí presentaremos el **tensor de inercia** que, para nuestros propósitos, puede definirse como una generalización del momento escalar de inercia de un objeto.

Ahora definiremos un conjunto de cantidades que proporcionan información sobre la distribución de la masa de un cuerpo rígido con respecto a una trama de referencia. La figura 6.1 muestra un cuerpo rígido con una trama adscrita. Pueden definirse los tensores de inercia en relación con cualquier trama, pero siempre consideraremos el

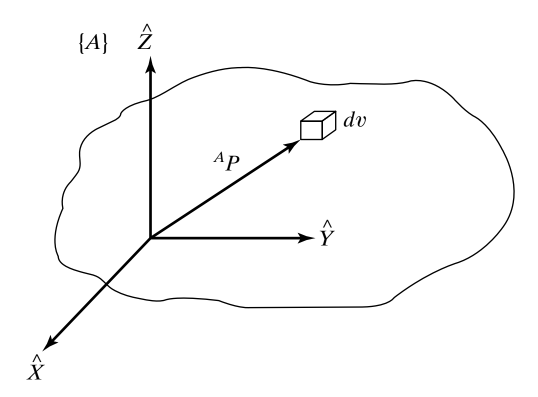


FIGURA 6.1: El tensor de inercia de un objeto describe la distribución de la masa de ese objeto. Aquí, el vector ${}^AP$ posiciona el elemento de volumen diferencial, dv.

caso de un tensor de inercia definido para una trama unida al cuerpo rígido. Cuando sea importante indicaremos, con un superíndice a la izquierda, la trama de referencia de un tensor de inercia dado. El tensor de inercia relativo a la trama $\{A\}$ se expresa en forma matricial como la siguiente matriz de 3×3:

$$
{}^AI = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}, \tag{6.16}
$$

en donde los elementos escalares están dados por

$$
\begin{aligned}
I_{xx} &= \iiint_V (y^2 + z^2)\rho dv, \\
I_{yy} &= \iiint_V (x^2 + z^2)\rho dv, \\
I_{zz} &= \iiint_V (x^2 + y^2)\rho dv, \\
I_{xy} &= \iiint_V xy\rho dv, \\
I_{xz} &= \iiint_V xz\rho dv, \\
I_{yz} &= \iiint_V yz\rho dv,
\end{aligned} \tag{6.17}
$$

aquí, el cuerpo rígido está compuesto de elementos de volumen diferencial, dv, que contienen material de densidad ρ. Cada elemento de volumen está ubicado con un vector, ${}^AP = [x\ y\ z]^T$, como se muestra en la figura 6.1.

Los elementos I_{xx}, I_{yy} e I_{zz} se llaman **momentos de inercia de masas**. Observe que, en cada caso estamos integrando los elementos de la masa, ρdv, multiplicados por los cuadrados de las distancias perpendiculares al eje correspondiente. Los elementos con índices mezclados se llaman **productos de inercia de masas**. Este conjunto de seis

cantidades independientes dependerá, para un cuerpo dado, de la posición y orientación de la trama en la que estén definidas. Si podemos elegir la orientación de la trama de referencia, es posible hacer que los productos de inercia sean cero. Cuando se alinean así los ejes de la trama de referencia se llaman **ejes principales**, y los momentos de masas correspondientes son los **momentos principales** de inercia.

EJEMPLO 6.1

Encuentre el tensor de inercia para el cuerpo rectangular de densidad ρ uniforme con respecto al sistema de coordenadas que se muestra en la figura 6.2.

Primero calculamos I_{xx}. Usando el elemento de volumen $dv = dx\,dy\,dz$, obtenemos

$$\begin{aligned} I_{xx} &= \int_0^h \int_0^l \int_0^\omega (y^2 + z^2)\rho\, dx\, dy\, dz \\ &= \int_0^h \int_0^l (y^2 + z^2)\omega\rho dy\, dz \\ &= \int_0^h \left(\frac{l^3}{3} + z^2 l\right)\omega\rho dz \\ &= \left(\frac{hl^3\omega}{3} + \frac{h^3 l\omega}{3}\right)\rho \\ &= \frac{m}{3}(l^2 + h^2), \end{aligned} \tag{6.18}$$

en donde m es la masa total del cuerpo. Si permutamos los términos podemos obtener I_{yy} e I_{zz} por inspección:

$$I_{yy} = \frac{m}{3}(\omega^2 + h^2) \tag{6.19}$$

y

$$I_{zz} = \frac{m}{3}(l^2 + \omega^2). \tag{6.20}$$

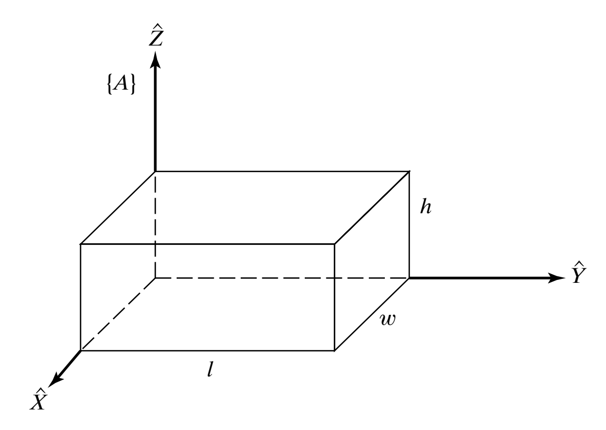


FIGURA 6.2: Un cuerpo de densidad uniforme.

A continuación calculamos I_{xy}:

$$\begin{aligned} I_{xy} &= \int_0^h \int_0^l \int_0^\omega xy\rho\, dx\, dy\, dz \\ &= \int_0^h \int_0^l \frac{\omega^2}{2} y\rho\, dy\, dz \\ &= \int_0^h \frac{\omega^2 l^2}{4} \rho dz \\ &= \frac{m}{4}\omega l. \end{aligned} \tag{6.21}$$

Si permutamos los términos obtenemos

$$I_{xz} = \frac{m}{4} h\omega \tag{6.22}$$

y

$$I_{yz} = \frac{m}{4} hl. \tag{6.23}$$

Por lo tanto, el tensor de inercia para este objeto es

$${}^A I = \begin{bmatrix} \frac{m}{3}(l^2 + h^2) & -\frac{m}{4}\omega l & -\frac{m}{4}h\omega \\ -\frac{m}{4}\omega l & \frac{m}{3}(\omega^2 + h^2) & -\frac{m}{4}hl \\ -\frac{m}{4}h\omega & -\frac{m}{4}hl & \frac{m}{3}(l^2 + \omega^2) \end{bmatrix}. \tag{6.24}$$

Como se indicó, el tensor de inercia es una función de la ubicación y la orientación de la trama de referencia. El **teorema de ejes paralelos** (un resultado bien conocido) es una manera de calcular cómo cambia el tensor de inercia bajo las *traslaciones* del sistema de coordenadas de referencia. El teorema de ejes paralelos relaciona el tensor de inercia en una trama cuyo origen está en el centro de masa, con el tensor de inercia con respecto a otra trama de referencia. Cuando $\{C\}$ se encuentra en el centro de masas del cuerpo y $\{A\}$ es una trama trasladada arbitrariamente, el teorema puede definirse [1] como

$$\begin{aligned} {}^A I_{zz} &= {}^C I_{zz} + m(x_c^2 + y_c^2), \\ {}^A I_{xy} &= {}^C I_{xy} - m x_c y_c, \end{aligned} \tag{6.25}$$

en donde $P_c = [x_c, y_c, z_c]^T$ ubica el centro de masas relativo a $\{A\}$; los momentos y productos de inercia restantes se calculan a partir de permutaciones de x, y y z en la ecuación (6.25). El teorema puede definirse en forma vectorial-matricial como

$${}^A I = {}^C I + m[P_c^T P_c I_3 - P_c P_c^T], \tag{6.26}$$

en donde I_3 es la matriz de identidad de 3×3.

EJEMPLO 6.2

Encuentre el tensor de inercia para el mismo cuerpo sólido descrito en el ejemplo 6.1, cuando está descrito en un sistema de coordenadas con el origen en el centro de masas del cuerpo.

Podemos aplicar el teorema de ejes paralelos (6.25), en donde

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \omega \\ l \\ h \end{bmatrix}.$$

Después encontramos

$$\begin{aligned} {}^{C}I_{zz} &= \frac{m}{12}(\omega^2 + l^2), \\ {}^{C}I_{xy} &= 0. \end{aligned} \tag{6.27}$$

Los demás elementos se encuentran por simetría. El tensor de inercia resultante escrito en la trama que está en el centro de masas es

$${}^{C}I = \begin{bmatrix} \frac{m}{12}(h^2 + l^2) & 0 & 0 \\ 0 & \frac{m}{12}(\omega^2 + h^2) & 0 \\ 0 & 0 & \frac{m}{12}(l^2 + \omega^2) \end{bmatrix}. \tag{6.28}$$

El resultado es diagonal, por lo que la trama $\{C\}$ debe representar los ejes principales de este cuerpo.

Algunos hechos adicionales sobre los tensores de inercia son:

1. Si dos ejes de la trama de referencia forman un plano de simetría para la distribución de la masa del cuerpo, los productos de inercia que tengan como índice la coordenada que sea normal al plano de simetría serán cero.
2. Los momentos de inercia siempre deben ser positivos. Los productos de inercia pueden tener cualquier signo.
3. La suma de los tres momentos de inercia está invariante bajo cambios de orientación en la trama de referencia.
4. Los valores propios de un tensor de inercia son los momentos principales para el cuerpo. Los vectores propios asociados son los ejes principales.

La mayoría de los manipuladores tienen vínculos cuya geometría y composición son algo complejas, de manera que es difícil en la práctica aplicar la ecuación (6.17). Una opción pragmática es medir, en lugar de calcular el momento de inercia de cada vínculo, utilizando un dispositivo de medición (por ejemplo, un *péndulo de inercia*).

6.4 ECUACIÓN DE NEWTON, ECUACIÓN DE EULER

Vamos a considerar cada vínculo de un manipulador como un cuerpo rígido. Si conocemos la ubicación del centro de masas y el tensor de inercia del vínculo, entonces su distribución de masa está completamente caracterizada. Para mover los vínculos debemos acelerarlos y desacelerarlos. Las fuerzas requeridas para dichos movimientos son una función de la aceleración deseada y de la distribución de la masa de los vínculos. La ecuación de Newton, junto con su análoga de rotación, la ecuación de Euler, describe la manera en que se relacionan las fuerzas, inercias y aceleraciones.

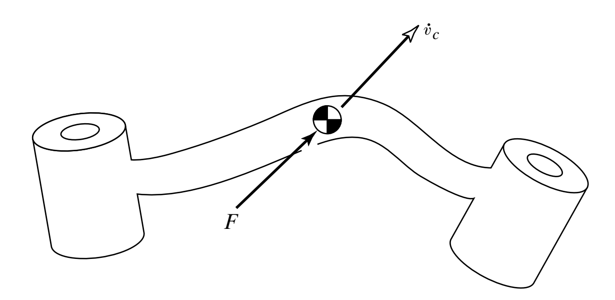


FIGURA 6.3: Una fuerza F que actúa en el centro de masas de un cuerpo hace que éste se acelere a $\dot{v}$.

Ecuación de Newton

La figura 6.3 muestra un cuerpo rígido cuyo centro de masas tiene una aceleración representada por $\dot{v}_C$. En dicha situación, la fuerza F que actúa en el centro de masas y produce esta aceleración se da mediante la ecuación de Newton:

$$F = m\dot{v}_C, \tag{6.29}$$

en donde m es la masa total del cuerpo.

Ecuación de Euler

La figura 6.4 muestra un cuerpo rígido que gira con una velocidad angular ω y una aceleración angular $\dot{\omega}$. En dicha situación, el momento N que debe estar actuando en el cuerpo para producir este movimiento, se obtiene mediante la ecuación de Euler:

$$N = {}^{C}I\dot{\omega} + \omega \times {}^{C}I\omega, \tag{6.30}$$

en donde ${}^{C}I$ es el tensor de inercia del cuerpo escrito en una trama $\{C\}$, cuyo origen se encuentra en el centro de masas.

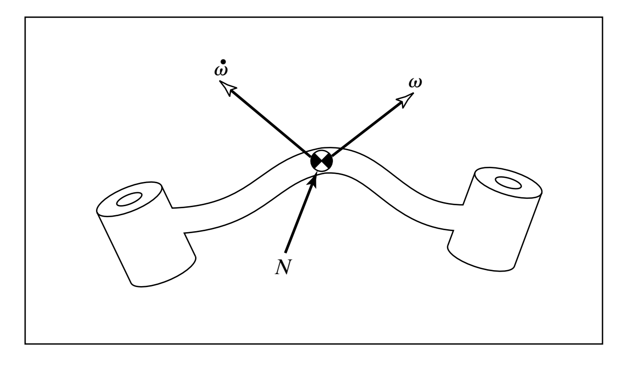


FIGURA 6.4: Un momento N que actúa sobre un cuerpo, y éste gira con una velocidad ω y acelera a $\dot{\omega}$.

6.5 FORMULACIÓN DINÁMICA ITERATIVA DE NEWTON-EULER

Ahora consideraremos el problema de calcular los momentos de torsión que corresponden a una trayectoria dada de un manipulador. Vamos a suponer que conocemos la posición, velocidad y aceleración de las articulaciones (Θ, $\dot{\Theta}$, $\ddot{\Theta}$). Con estos datos, y conociendo la información de cinemática y distribución de masa del robot, podemos calcular los momentos de torsión de las articulaciones requeridos para producir este movimiento. El algoritmo que se presenta está basado en el método publicado por Luh, Walter y Paul en [2].

Iteraciones salientes para calcular velocidades y aceleraciones

Para calcular las fuerzas inerciales que actúan en los vínculos es necesario calcular la velocidad de rotación y la aceleración lineal y rotacional del centro de masas de cada vínculo del manipulador en cualquier instante dado. Estos cálculos deben realizarse en forma iterativa, empezando con el vínculo 1 y avanzando sucesivamente, vínculo por vínculo *hacia fuera* hasta el vínculo n.

En el capítulo 5 hablamos sobre la "propagación" de velocidad de rotación de un vínculo a otro, y se da (para la articulación giratoria $i + 1$) mediante

$$ {}^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\, {}^{i}\omega_i + \dot{\theta}_{i+1}\, {}^{i+1}\hat{Z}_{i+1}. \tag{6.31}$$

De (6.15) obtenemos la ecuación para transformar la aceleración angular de un vínculo al siguiente:

$$ {}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}R\, {}^{i}\dot{\omega}_i + {}^{i+1}_{i}R\, {}^{i}\omega_i \times \dot{\theta}_{i+1}\, {}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1}\, {}^{i+1}\hat{Z}_{i+1}. \tag{6.32}$$

Cuando la articulación $i + 1$ es prismática, esto se simplifica a

$$ {}^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}R\, {}^{i}\omega_i. \tag{6.33}$$

La aceleración lineal del origen de cada trama de vínculo se obtiene mediante la aplicación de la ecuación (6.12):

$$ {}^{i+1}\dot{v}_{i+1} = {}^{i+1}_{i}R[{}^{i}\omega_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{i+1}) + {}^{i}\dot{v}_i], \tag{6.34}$$

Para la articulación $i + 1$ prismática, la ecuación (6.34) se convierte [de la ecuación (6.10)] en

$$ \begin{aligned} {}^{i+1}\dot{v}_{i+1} = {}^{i+1}_{i}R({}^{i}\dot{\omega}_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{i+1}) + {}^{i}\dot{v}_i) \\ + 2\, {}^{i+1}\omega_{i+1} \times \dot{d}_{i+1}\, {}^{i+1}\hat{Z}_{i+1} + \ddot{d}_{i+1}\, {}^{i+1}\hat{Z}_{i+1}. \end{aligned} \tag{6.35}$$

Asimismo, necesitaremos la aceleración lineal del centro de masas de cada vínculo, que también puede encontrarse aplicando la ecuación (6.12):

$$ {}^{i}\dot{v}_{C_i} = {}^{i}\dot{\omega}_i \times {}^{i}P_{C_i} + {}^{i}\omega_i \times ({}^{i}\omega_i + {}^{i}P_{C_i}) + {}^{i}\dot{v}_i, \tag{6.36}$$

Aquí imaginamos una trama $\{C_i\}$ unida a cada vínculo, con su origen ubicado en el centro de masas del vínculo y con la misma orientación que la trama del vínculo, $\{i\}$. La ecuación (6.36) no implica el movimiento de la articulación, por lo cual es válida para la articulación $i + 1$, sin importar que sea angular o prismática.

Observe que la aplicación de las ecuaciones al vínculo 1 es especialmente simple, ya que ${}^{0}\omega_0 = {}^{0}\dot{\omega}_0 = 0$.

La fuerza y el momento de torsión que actúan sobre un vínculo

Después de calcular las aceleraciones lineal y angular del centro de masas de cada vínculo, podemos aplicar las ecuaciones de Newton-Euler (sección 6.4) para calcular la fuerza y el momento de torsión inerciales que actúan en el centro de masas de cada vínculo. Por lo tanto tenemos que

$$F_i = m\dot{v}_{C_i},$$
$$N_i = {}^{C_i}I\dot{\omega}_i + \omega_i \times {}^{C_i}I\omega_i, \tag{6.37}$$

en donde $\{C_i\}$ tiene su origen en el centro de masas del vínculo y la misma orientación que la trama del vínculo, $\{i\}$.

Iteraciones entrantes para calcular fuerzas y momentos de torsión

Si ya calculamos las fuerzas y momentos de torsión que actúan en cada vínculo, ahora necesitamos calcular los momentos de torsión de articulación que producirán estas fuerzas y los momentos de torsión netos que se aplican a cada vínculo.

Podemos hacer esto escribiendo una ecuación de balance de fuerza y balance de momento, basada en el diagrama de cuerpo libre de un vínculo común. (Vea la figura 6.5). Sobre cada vínculo se ejercen fuerzas y momentos de torsión provenientes de los vínculos adyacentes, además de que experimenta una fuerza y un momento de torsión inerciales. En el capítulo 5 definimos símbolos especiales para la fuerza y el momento de torsión ejercidos por un vínculo adyacente, los cuales repetiremos aquí:

f_i = fuerza ejercida en el vínculo i por el vínculo $i - 1$,

n_i = momento de torsión ejercido en el vínculo i por el vínculo $i - 1$.

Al sumar las fuerzas que actúan sobre el vínculo i llegamos a la relación de balanceo de fuerzas:

$${}^iF_i = {}^if_i - {}^{i}_{i+1}R\,{}^{i+1}f_{i+1}. \tag{6.38}$$

Al sumar los momentos de torsión sobre el centro de masas e igualarlos a cero, llegamos a la ecuación de balanceo de momentos de torsión:

$${}^iN_i = {}^in_i - {}^in_{i+1} + (-{}^iP_{C_i}) \times {}^if_i - ({}^iP_{i+1} - {}^iP_{C_i}) \times {}^if_{i+1}. \tag{6.39}$$

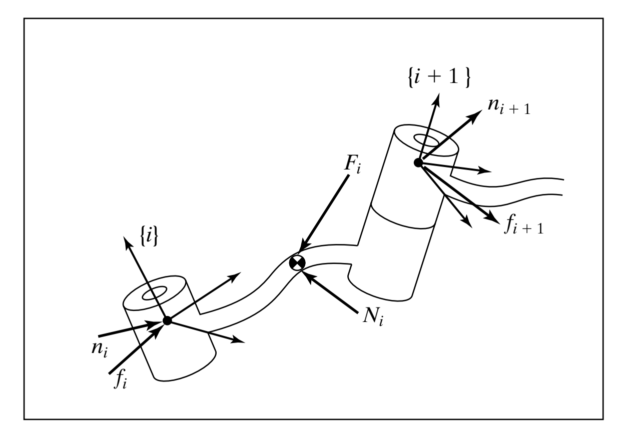


FIGURA 6.5: El balanceo de fuerzas, incluyendo las fuerzas inerciales, para el vínculo de un manipulador.

Si utilizamos el resultado de la relación de balanceo de fuerzas (6.38) y agregamos unas cuantas matrices de rotación, podemos escribir la ecuación (6.39) como

$$ {}^{i}N_i = {}^{i}n_i - {}^{i}_{i+1}R\, {}^{i+1}n_{i+1} - {}^{i}P_{C_i} \times {}^{i}F_i - {}^{i}P_{i+1} \times {}^{i}_{i+1}R\, {}^{i+1}f_{i+1}. \tag{6.40} $$

Finalmente, podemos reordenar las ecuaciones de fuerza y momento de torsión para que aparezcan como relaciones iterativas desde el vínculo adyacente de mayor numeración hasta el de menor numeración:

$$ {}^{i}f_i = {}^{i}_{i+1}R\, {}^{i+1}f_{i+1} + {}^{i}F_i, \tag{6.41} $$

$$ {}^{i}n_i = {}^{i}N_i + {}^{i}_{i+1}R\, {}^{i+1}n_{i+1} + {}^{i}P_{C_i} \times {}^{i}F_i + {}^{i}P_{i+1} \times {}^{i}_{i+1}R\, {}^{i+1}f_{i+1}. \tag{6.42} $$

Estas ecuaciones se evalúan vínculo por vínculo, empezando por n y trabajando hacia la base del robot. Estas *iteraciones entrantes de fuerza* son análogas a las iteraciones de fuerza estáticas presentadas en el capítulo 5, excepto que ahora se toman en cuenta las fuerzas y momentos de torsión inerciales en cada vínculo.

Como en el caso estático, los momentos de torsión de articulación requeridos pueden obtenerse tomando el componente $\hat{Z}$ del momento de torsión aplicado por un vínculo sobre su vínculo adyacente:

$$ \tau_i = {}^{i}n_i^{T}\, {}^{i}\hat{Z}_i. \tag{6.43} $$

Para la articulación i prismática utilizamos

$$ \tau_i = {}^{i}f_i^{T}\, {}^{i}\hat{Z}_i, \tag{6.44} $$

en donde hemos usado el símbolo τ para la fuerza de un actuador lineal.

Observe que, para un robot que se mueve en el espacio libre, ${}^{N+1}f_{N+1}$ y ${}^{N+1}n_{N+1}$ se hacen iguales a cero, por lo que la primera aplicación de las ecuaciones para el vínculo n es bastante simple. Si el robot está en contacto con el entorno, las fuerzas y momentos de torsión debidos a este contacto pueden incluirse en el balance de fuerzas haciendo que ${}^{N+1}f_{N+1}$ y ${}^{N+1}n_{N+1}$ sean distintos de cero.

El algoritmo iterativo de dinámica Newton-Euler

El algoritmo completo para calcular momentos de torsión de articulación a partir del movimiento de éstas, se compone de dos partes. En primer lugar, las velocidades y aceleraciones de los vínculos se calculan en forma iterativa desde el vínculo 1, hacia el vínculo n, y se aplican las ecuaciones de Newton-Euler en cada caso. En segundo lugar, las fuerzas y los momentos de torsión de interacción y los momentos de torsión del actuador de una articulación se calculan en forma recursiva, desde el vínculo n hasta el vínculo 1. A continuación, se muestra un resumen de las ecuaciones para el caso en que todas las articulaciones son giratorias:

Iteraciones salientes: i: $0 \rightarrow 5$

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\,{}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}, \tag{6.45}$$

$$^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}R\,{}^{i}\dot{\omega}_i + {}^{i+1}_{i}R\,{}^{i}\omega_i \times \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}, \tag{6.46}$$

$$^{i+1}\dot{v}_{i+1} = {}^{i+1}_{i}R({}^{i}\dot{\omega}_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{i+1}) + {}^{i}\dot{v}_i), \tag{6.47}$$

$$\begin{aligned} ^{i+1}\dot{v}_{C_{i+1}} &= {}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{C_{i+1}} \\ &\quad + {}^{i+1}\omega_{i+1} \times ({}^{i+1}\omega_{i+1} \times {}^{i+1}P_{C_{i+1}}) + {}^{i+1}\dot{v}_{i+1}, \end{aligned} \tag{6.48}$$

$$^{i+1}F_{i+1} = m_{i+1}\,{}^{i+1}\dot{v}_{C_{i+1}}, \tag{6.49}$$

$$^{i+1}N_{i+1} = {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\omega_{i+1}. \tag{6.50}$$

Iteraciones entrantes: i : $6 \rightarrow 1$

$$^{i}f_i = {}^{i}_{i+1}R\,{}^{i+1}f_{i+1} + {}^{i}F_i, \tag{6.51}$$

$$\begin{aligned} ^{i}n_i &= {}^{i}N_i + {}^{i}_{i+1}R\,{}^{i+1}n_{i+1} + {}^{i}P_{C_i} \times {}^{i}F_i \\ &\quad + {}^{i}P_{i+1} \times {}^{i}_{i+1}R\,{}^{i+1}f_{i+1}, \end{aligned} \tag{6.52}$$

$$\tau_i = {}^{i}n_i^T\,{}^{i}\hat{Z}_i. \tag{6.53}$$

Inclusión de fuerzas de gravedad en el algoritmo de dinámica

El efecto de carga de la gravedad sobre los vínculos puede incluirse muy fácilmente haciendo que ${}^{0}\dot{v}_0 = G$, en donde G tiene la magnitud del vector de gravedad pero apunta en la dirección opuesta. Esto es equivalente a decir que la base del robot está acelerando hacia arriba con una aceleración de 1 g. Esta aceleración ficticia hacia arriba produce exactamente el mismo efecto en los vínculos que tendría la gravedad. Así se calcula el efecto de la gravedad sin necesidad de incurrir en un gasto computacional adicional.

6.6 COMPARACIÓN ENTRE FORMA ITERATIVA Y FORMA CERRADA

Las ecuaciones (6.46) a (6.53) proporcionan un esquema computacional con el que, dadas las posiciones de articulaciones, velocidades y aceleraciones, podemos calcular los momentos de torsión de articulación requeridos. Al igual que con las ecuaciones que desarrollamos para calcular el jacobiano en el capítulo 5, estas relaciones pueden usarse de dos formas: como un algoritmo computacional numérico o como un algoritmo usado analíticamente para desarrollar ecuaciones simbólicas.

Es atractivo usar las ecuaciones como un algoritmo computacional numérico, ya que éstas se aplican a cualquier robot. Una vez que se especifican los tensores de inercia, las masas de los vínculos, los vectores P_{Ci} y las matrices ${}^{i+1}_{i}R$ para un manipulador específico, las ecuaciones pueden aplicarse directamente para calcular los momentos de torsión de articulación correspondientes a cualquier movimiento.

No obstante, es conveniente tener una comprensión más detallada de la estructura de las ecuaciones; por ejemplo, ¿cuál es la forma de los términos de gravedad? ¿Cómo se compara la magnitud de los efectos de la gravedad con la magnitud de los efectos inerciales? Para investigar éstas y otras preguntas, a menudo es conveniente escribir ecuaciones dinámicas de forma cerrada. Estas ecuaciones pueden derivarse aplicando

las ecuaciones recursivas de Newton-Euler simbólicamente a Θ, $\dot{\Theta}$ y $\ddot{\Theta}$; lo cual es análogo a lo que hicimos en el capítulo 5 para derivar la forma simbólica del jacobiano.

6.7 UN EJEMPLO DE ECUACIONES DINÁMICAS DE FORMA CERRADA

Ahora calcularemos las ecuaciones dinámicas de forma cerrada para el manipulador planar de dos vínculos que se muestra en la figura 6.6. Por sencillez supondremos que la distribución de la masa es extremadamente simple: toda la masa existe como una masa puntual en el extremo distal de cada vínculo. Estas masas son m_1 y m_2.

En primer lugar, vamos a determinar los valores de las diversas cantidades que aparecerán en las ecuaciones recursivas de Newton-Euler. Los vectores que ubican el centro de masas para cada vínculo son

$$^{1}P_{C_1} = l_1\hat{X}_1,$$
$$^{2}P_{C_2} = l_2\hat{X}_2.$$

Como se supone una masa puntual, el tensor de inercia escrito en el centro de masas para cada vínculo es la matriz cero:

$$^{C_1}I_1 = 0,$$
$$^{C_2}I_2 = 0.$$

Como no hay fuerzas actuando en el efector final, tenemos que

$$f_3 = 0,$$
$$n_3 = 0.$$

La base del robot no está girando; por lo tanto,

$$\omega_0 = 0,$$
$$\dot{\omega}_0 = 0.$$

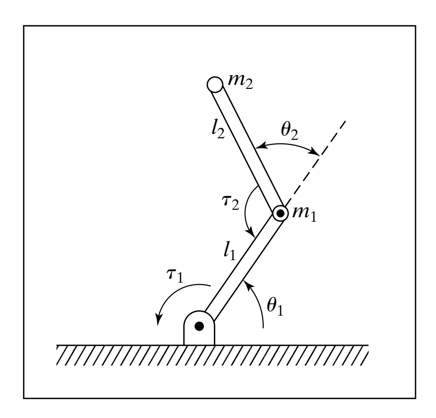


FIGURA 6.6: Manipulador planar de dos vínculos con masas puntuales en los extremos distales de los vínculos.

Para incluir las fuerzas de gravedad, utilizaremos

$$ {}^{0}\dot{v}_0 = g\hat{Y}_0. $$

La rotación entre tramas de vínculo sucesivas está dada por

$$ {}^{i}_{i+1}R = \begin{bmatrix} c_{i+1} & -s_{i+1} & 0.0 \\ s_{i+1} & c_{i+1} & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}, $$

$$ {}^{i+1}_{i}R = \begin{bmatrix} c_{i+1} & s_{i+1} & 0.0 \\ -s_{i+1} & c_{i+1} & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}. $$

Ahora vamos a aplicar las ecuaciones (6.46) a (6.53).

Las iteraciones salientes para el vínculo 1 son:

$$ \begin{aligned} {}^{1}\omega_1 &= \dot{\theta}_1\,{}^{1}\hat{Z}_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix}, \\ {}^{1}\dot{\omega}_1 &= \ddot{\theta}_1\,{}^{1}\hat{Z}_1 = \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{bmatrix}, \\ {}^{1}\dot{v}_1 &= \begin{bmatrix} c_1 & s_1 & 0 \\ -s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} = \begin{bmatrix} gs_1 \\ gc_1 \\ 0 \end{bmatrix}, \\ {}^{1}\dot{v}_{C_1} &= \begin{bmatrix} 0 \\ l_1\ddot{\theta}_1 \\ 0 \end{bmatrix} + \begin{bmatrix} -l_1\dot{\theta}_1^2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} gs_1 \\ gc_1 \\ 0 \end{bmatrix} = \begin{bmatrix} -l_1\dot{\theta}_1^2 + gs_1 \\ l_1\ddot{\theta}_1 + gc_1 \\ 0 \end{bmatrix}, \\ {}^{1}F_1 &= \begin{bmatrix} -m_1 l_1\dot{\theta}_1^2 + m_1 g s_1 \\ m_1 l_1\ddot{\theta}_1 + m_1 g c_1 \\ 0 \end{bmatrix}, \\ {}^{1}N_1 &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \end{aligned} \tag{6.54} $$

Las iteraciones salientes para el vínculo 2 son:

$$ {}^{2}\omega_2 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix}, $$

$$ {}^{2}\dot{\omega}_2 = \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 + \ddot{\theta}_2 \end{bmatrix}, $$

$$
{}^2\dot{v}_2 = \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -l_1\dot{\theta}_1^2 + gs_1 \\ l_1\ddot{\theta}_1 + gc_1 \\ 0 \end{bmatrix} = \begin{bmatrix} l_1\ddot{\theta}_1 s_2 - l_1\dot{\theta}_1^2 c_2 + gs_{12} \\ l_1\ddot{\theta}_1 c_2 + l_1\dot{\theta}_1^2 s_2 + gc_{12} \\ 0 \end{bmatrix},
$$

$$
{}^2\dot{v}_{C_2} = \begin{bmatrix} 0 \\ l_2(\ddot{\theta}_1 + \ddot{\theta}_2) \\ 0 \end{bmatrix} + \begin{bmatrix} -l_2(\dot{\theta}_1 + \dot{\theta}_2)^2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} l_1\ddot{\theta}_1 s_2 - l_1\dot{\theta}_1^2 c_2 + gs_{12} \\ l_1\ddot{\theta}_1 c_2 + l_1\dot{\theta}_1^2 s_2 + gc_{12} \\ 0 \end{bmatrix}, \tag{6.55}
$$

$$
{}^2F_2 = \begin{bmatrix} m_2 l_1\ddot{\theta}_1 s_2 - m_2 l_1\dot{\theta}_1^2 c_2 + m_2 g s_{12} - m_2 l_2(\dot{\theta}_1 + \dot{\theta}_2)^2 \\ m_2 l_1\ddot{\theta}_1 c_2 + m_2 l_1\dot{\theta}_1^2 s_2 + m_2 g c_{12} + m_2 l_2(\ddot{\theta}_1 + \ddot{\theta}_2) \\ 0 \end{bmatrix},
$$

$$
{}^2N_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.
$$

Las iteraciones entrantes para el vínculo 2 son:

$$
{}^2f_2 = {}^2F_2,
$$

$$
{}^2n_2 = \begin{bmatrix} 0 \\ 0 \\ m_2 l_1 l_2 c_2\ddot{\theta}_1 + m_2 l_1 l_2 s_2\dot{\theta}_1^2 + m_2 l_2 g c_{12} + m_2 l_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) \end{bmatrix}. \tag{6.56}
$$

Las iteraciones entrantes para el vínculo 1 son:

$$
{}^1f_1 = \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m_2 l_1 s_2\ddot{\theta}_1 - m_2 l_1 c_2\dot{\theta}_1^2 + m_2 g s_{12} - m_2 l_2(\dot{\theta}_1 + \dot{\theta}_2)^2 \\ m_2 l_1 c_2\ddot{\theta}_1 + m_2 l_1 s_2\dot{\theta}_1^2 + m_2 g c_{12} + m_2 l_2(\ddot{\theta}_1 + \ddot{\theta}_2) \\ 0 \end{bmatrix} + \begin{bmatrix} -m_1 l_1\dot{\theta}_1^2 + m_1 g s_1 \\ m_1 l_1\ddot{\theta}_1 + m_1 g c_1 \\ 0 \end{bmatrix},
$$

$$
{}^1n_1 = \begin{bmatrix} 0 \\ 0 \\ m_2 l_1 l_2 c_2\ddot{\theta}_1 + m_2 l_1 l_2 s_2\dot{\theta}_1^2 + m_2 l_2 g c_{12} + m_2 l_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ m_1 l_1^2\ddot{\theta}_1 + m_1 l_1 g c_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ m_2 l_1^2\ddot{\theta}_1 - m_2 l_1 l_2 s_2(\dot{\theta}_1 + \dot{\theta}_2)^2 + m_2 l_1 g s_2 s_{12} \\ + m_2 l_1 l_2 c_2(\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 g c_2 c_{12} \end{bmatrix}. \tag{6.57}
$$

Para encontrar los momentos de torsión extraemos los componentes $\hat{Z}$ de las ${}^i n_i$:

$$\tau_1 = m_2 l_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) + (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 s_2 \dot{\theta}_2^2$$
$$-2 m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1,$$
$$\tau_2 = m_2 l_1 l_2 c_2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + m_2 l_2 g c_{12} + m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2). \tag{6.58}$$

Las ecuaciones (6.58) proporcionan expresiones para el momento de torsión en los actuadores como una función de la posición de la articulación, su velocidad y su aceleración. Tenga en cuenta que estas funciones algo complejas surgieron de uno de los manipuladores más simples que pueda imaginarse. Obviamente, las ecuaciones de forma cerrada para un manipulador con seis grados de libertad serán considerablemente más complejas.

6.8 LA ESTRUCTURA DE LAS ECUACIONES DINÁMICAS DE UN MANIPULADOR

A menudo es conveniente expresar las ecuaciones dinámicas de un manipulador en una sola ecuación que oculte algunos de los detalles, pero que muestre parte de la estructura de las ecuaciones.

La ecuación en el espacio de estados

Cuando las ecuaciones de Newton-Euler se evalúan simbólicamente para cualquier manipulador, producen una ecuación dinámica que puede escribirse de la siguiente forma

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta), \tag{6.59}$$

en donde $M(\Theta)$ es la **matriz de masas** de $n \times n$ del manipulador, $V(\Theta, \dot{\Theta})$ es un vector de $n \times 1$, de términos centrífugos y Coriolis, y $G(\Theta)$ es un vector de $n \times 1$, de términos de gravedad. Utilizaremos el término **ecuación en el espacio de estados** porque el término $V(\Theta, \dot{\Theta})$, que aparece en la ecuación (6.59) tiene dependencias tanto de posición como de velocidad [3].

Cada elemento de $M(\Theta)$ y de $G(\Theta)$ es una función compleja que depende de Θ, la posición de todas las articulaciones del manipulador. Cada elemento de $V(\Theta, \dot{\Theta})$ es una función compleja tanto de Θ como de $\dot{\Theta}$.

Podemos separar los diversos tipos de términos que aparecen en las ecuaciones dinámicas y formar la matriz de masas del manipulador, el vector centrífugo y de Coriolis, y el vector de gravedad.

EJEMPLO 6.3

Proporcione los valores de $M(\Theta)$, $V(\Theta, \dot{\Theta})$ y $G(\Theta)$ para el manipulador de la sección 6.7.

La ecuación (6.59) define la matriz de masas del manipulador, $M(\Theta)$; está compuesta de todos los términos que multiplican a $\ddot{\Theta}$ y es una función de Θ. Por lo tanto, tenemos que

$$M(\Theta) = \begin{bmatrix} l_2^2 m_2 + 2 l_1 l_2 m_2 c_2 + l_1^2 (m_1 + m_2) & l_2^2 m_2 + l_1 l_2 m_2 c_2 \\ l_2^2 m_2 + l_1 l_2 m_2 c_2 & l_2^2 m_2 \end{bmatrix}. \tag{6.60}$$

Cualquier matriz de masas de un manipulador es simétrica y está definida como positiva; por lo tanto, siempre es invertible.

El término velocidad $V(\Theta, \dot{\Theta})$ contiene todos los elementos de cualquier dependencia en la velocidad de las articulaciones. Por lo tanto,

$$V(\Theta, \dot{\Theta}) = \begin{bmatrix} -m_2 l_1 l_2 s_2 \dot{\theta}_2^2 - 2 m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 \\ m_2 l_1 l_2 s_2 \dot{\theta}_1^2 \end{bmatrix}. \tag{6.61}$$

Un término como $-m_2 l_1 l_2 s_2 \dot{\theta}_2^2$ se produce por una **fuerza centrífuga**, y se reconoce como tal, debido a que depende de la raíz cuadrada de una velocidad de articulación. Un término como $-2m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2$ se produce por una **fuerza de Coriolis** y siempre contiene el producto de dos velocidades de articulación distintas.

El término de gravedad $G(\Theta)$ contiene todos los términos en los que aparece la constante gravitacional g. Por lo tanto, tenemos que

$$G(\Theta) = \begin{bmatrix} m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1 \\ m_2 l_2 g c_{12} \end{bmatrix}. \tag{6.62}$$

Tenga en cuenta que el término de gravedad depende solamente de Θ y no de sus derivadas.

La ecuación en el espacio de configuración

Si escribimos el término dependiente de la velocidad $V(\Theta, \dot{\Theta})$ en una forma distinta, podemos escribir las ecuaciones dinámicas así:

$$\tau = M(\Theta)\ddot{\Theta} + B(\Theta)[\dot{\Theta}\dot{\Theta}] + C(\Theta)[\dot{\Theta}^2] + G(\Theta), \tag{6.63}$$

en donde $B(\Theta)$ es una matriz de dimensiones $n \times n(n-1)/2$ de coeficientes de Coriolis; $[\dot{\Theta}\,\dot{\Theta}]$ es un vector de $n(n-1)/2 \times 1$ de productos de velocidad de articulación dados por

$$[\dot{\Theta}\dot{\Theta}] = [\dot{\theta}_1\dot{\theta}_2\ \dot{\theta}_1\dot{\theta}_3\ \ldots\ \dot{\theta}_{n-1}\dot{\theta}_n]^T, \tag{6.64}$$

$C(\Theta)$ es una matriz de $n \times n$ de coeficientes centrífugos y $[\dot{\Theta}^2]$ es un vector de $n \times 1$ dado por

$$[\dot{\theta}_1^2\ \dot{\theta}_2^2 \ldots \dot{\theta}_n^2]^T. \tag{6.65}$$

Llamaremos a (6.63) la **ecuación en el espacio de configuración**, ya que las matrices son solamente funciones de la posición del manipulador [3].

En esta forma de las ecuaciones dinámicas, la complejidad del cálculo está en la forma de calcular varios parámetros que son solamente una función de la posición del manipulador (Θ). Esto es importante en aplicaciones (como el control computacional de un manipulador) en las que las ecuaciones dinámicas deben actualizarse a medida que se mueve el manipulador. [La ecuación (6.63) proporciona una forma en la que los parámetros son una función de la posición de articulación solamente y pueden actualizarse en una proporción relacionada con la rapidez con la que el manipulador cambia de configuración]. En el capítulo 10 consideraremos de nuevo esta forma cuando veamos el problema del control del manipulador.

EJEMPLO 6.4

Proporcione los valores de $B(\Theta)$ y $C(\Theta)$ [de la ecuación (6.63)] para el manipulador de la sección 6.7.

Para este manipulador simple de dos vínculos tenemos que

$$[\dot{\Theta}\Theta] = [\dot{\theta}_1\dot{\theta}_2],$$

$$[\dot{\Theta}^2] = \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \end{bmatrix}. \tag{6.66}$$

Por lo que podemos ver que

$$B(\Theta) = \begin{bmatrix} -2m_2 l_1 l_2 s_2 \\ 0 \end{bmatrix} \tag{6.67}$$

y

$$C(\Theta) = \begin{bmatrix} 0 & -m_2 l_1 l_2 s_2 \\ m_2 l_1 l_2 s_2 & 0 \end{bmatrix}. \tag{6.68}$$

6.9 FORMULACIÓN LAGRANGIANA DE LA DINÁMICA DE MANIPULADORES

El método de Newton-Euler se basa en las fórmulas dinámicas elementales (6.29) y (6.30) y en el análisis de fuerzas y momentos de restricción que actúan entre los vínculos. Como una alternativa para el método de Newton-Euler, en esta sección presentaremos brevemente la **formulación dinámica lagrangiana**. Podría decirse que la formulación de Newton-Euler es un método de "balance de fuerzas" dinámicas, y que la formulación lagrangiana es un método de "balance de energías" de la dinámica. Desde luego, para el mismo manipulador, ambos métodos proporcionarán las mismas ecuaciones de movimiento. Nuestra definición de la dinámica lagrangiana será breve y algo especializada en el caso de un manipulador mecánico de cadena serial con vínculos rígidos. Para obtener una referencia más completa y general, vea [4].

Empezaremos desarrollando una expresión para la energía cinética de un manipulador. La energía cinética del i-ésimo vínculo, que llamaremos k_i, puede expresarse como

$$k_i = \frac{1}{2} m_i v_{C_i}^T v_{C_i} + \frac{1}{2} \, {}^i\omega_i^T \, {}^{C_i}I_i \, {}^i\omega_i, \tag{6.69}$$

en donde el primer término es la energía cinética debida a la velocidad lineal del centro de masas del vínculo, y el segundo término es la energía cinética debida a la velocidad angular del vínculo. La energía cinética total del manipulador es la suma de la energía cinética en los vínculos individuales; esto es:

$$k = \sum_{i=1}^{n} k_i. \tag{6.70}$$

Las v_{C_i} e ${}^i\omega_i$ en la ecuación (6.69) son funciones de Θ y de $\dot{\Theta}$, por lo que podemos ver que la energía cinética de un manipulador puede describirse mediante una fórmula escalar como una función de la posición de articulación y la velocidad, $k(\Theta, \dot{\Theta})$. De hecho, la energía cinética de un manipulador se da mediante

$$k(\Theta, \dot{\Theta}) = \frac{1}{2} \dot{\Theta}^T M(\Theta) \dot{\Theta}, \tag{6.71}$$

en donde $M(\Theta)$ es la matriz de masas de $n \times n$ del manipulador, que ya presentamos en la sección 6.8. Una expresión con la forma de la ecuación (6.71) se conoce como **forma cuadrática** [5], ya que cuando se expande, la ecuación escalar resultante está compuesta únicamente de términos cuya dependencia sobre los $\dot{\theta}_i$ es cuadrática. Además, como la energía cinética total debe ser siempre positiva, la matriz de masas del manipulador debe ser una matriz **definida positiva**. Este tipo de matrices tienen la propiedad de que su forma cuadrática siempre es un escalar positivo. La ecuación (6.71) puede verse como análoga a la expresión familiar para la energía cinética de una masa puntual:

$$k = \frac{1}{2} m v^2. \tag{6.72}$$

El hecho de que la matriz de masas de un manipulador deba ser definida positiva, es análogo al hecho de que una masa escalar sea siempre un número positivo.

La energía potencial del i-ésimo vínculo, que llamaremos u_i, puede expresarse como

$$u_i = -m_i \, {}^0g^T \, {}^0P_{C_i} + u_{ref_i}, \tag{6.73}$$

en donde ${}^0g$ es el vector de gravedad de 3 × 1; ${}^0P_{C_i}$ es el vector que ubica al centro de masas del i-ésimo vínculo, y u_{ref_i} es una constante que se elige de manera que el valor mínimo de u_i sea cero.[1] La energía potencial total almacenada en el manipulador es la suma de la energía potencial en los vínculos individuales; esto es:

$$u = \sum_{i=1}^{n} u_i. \tag{6.74}$$

Como las ${}^0P_{C_i}$ en la ecuación (6.73) son funciones de Θ, podemos ver que la energía potencial de un manipulador puede describirse mediante una fórmula escalar como una función de la posición de articulación, $u(\Theta)$.

La formulación dinámica lagrangiana provee los medios para derivar las ecuaciones de movimiento a partir de una función escalar llamada **lagrangiano**, que se define como la diferencia entre la energía cinética y potencial de un sistema mecánico. En nuestra notación, el lagrangiano de un manipulador es

$$\mathcal{L}(\Theta, \dot{\Theta}) = k(\Theta, \dot{\Theta}) - u(\Theta). \tag{6.75}$$

Y las ecuaciones de movimiento para el manipulador se dan entonces así:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\Theta}} - \frac{\partial \mathcal{L}}{\partial \Theta} = \tau, \tag{6.76}$$

en donde τ es el vector de $n \times 1$ de momentos de torsión de los actuadores. En el caso de un manipulador, esta ecuación se convierte en

$$\frac{d}{dt}\frac{\partial k}{\partial \dot{\Theta}} - \frac{\partial k}{\partial \Theta} + \frac{\partial u}{\partial \Theta} = \tau, \tag{6.77}$$

en donde por brevedad hemos omitido los argumentos de $k(\cdot)$ y $u(\cdot)$.

EJEMPLO 6.5

Los vínculos de un manipulador RP, que se muestran en la figura 6.7, tienen los siguientes tensores de inercia:

$$ {}^{C_1}I_1 = \begin{bmatrix} I_{xx1} & 0 & 0 \\ 0 & I_{yy1} & 0 \\ 0 & 0 & I_{zz1} \end{bmatrix}, $$

$$ {}^{C_2}I_2 = \begin{bmatrix} I_{xx2} & 0 & 0 \\ 0 & I_{yy2} & 0 \\ 0 & 0 & I_{zz2} \end{bmatrix}, \tag{6.78}$$

[1] En realidad, sólo aparecerá en los cálculos dinámicos la derivada parcial de la energía potencial con respecto a Θ, por lo que esta constante es arbitraria. Esto corresponde a definir la energía potencial relativa a una altura de referencia arbitraria con valor de cero.

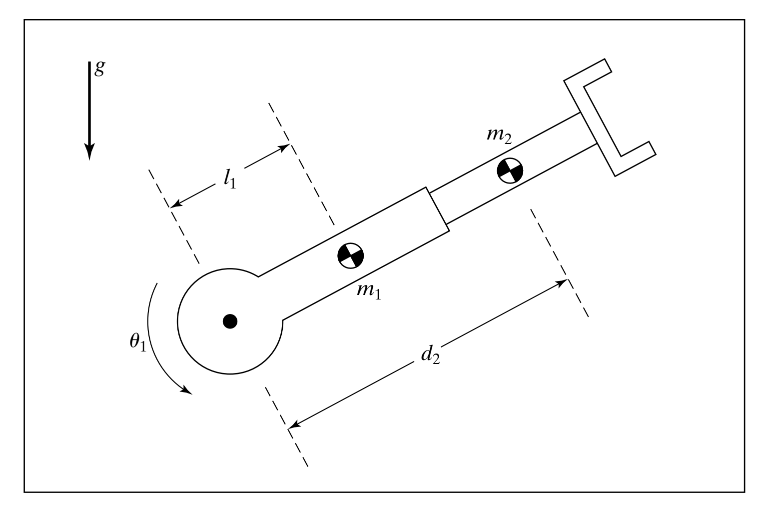


FIGURA 6.7: El manipulador RP del ejemplo 6.5.

y las masas totales m_1 y m_2. Como se muestra en la figura 6.7, el centro de masas del vínculo 1 se encuentra a una distancia l_1 del eje de la articulación 1, el centro de masas del vínculo 2 se encuentra a una distancia variable d_2 del eje de la articulación 1. Use dinámica lagrangiana para determinar la ecuación de movimiento para este manipulador.

Utilizando la ecuación (6.69), podemos escribir la energía cinética del vínculo 1 como:

$$k_1 = \frac{1}{2}m_1 l_1^2 \dot{\theta}_1^2 + \frac{1}{2}I_{zz1}\,\dot{\theta}_1^2 \tag{6.79}$$

y la energía cinética del vínculo 2 como

$$k_2 = \frac{1}{2}m_2(d_2^2\dot{\theta}_1^2 + \dot{d}_2^2) + \frac{1}{2}I_{zz2}\,\dot{\theta}_1^2. \tag{6.80}$$

Por lo tanto, la energía cinética total se obtiene mediante

$$k(\Theta, \dot{\Theta}) = \frac{1}{2}(m_1 l_1^2 + I_{zz1} + I_{zz2} + m_2 d_2^2)\dot{\theta}_1^2 + \frac{1}{2}m_2\dot{d}_2^2. \tag{6.81}$$

Si utilizamos la ecuación (6.73), podemos escribir la energía potencial del vínculo 1 como

$$u_1 = m_1 l_1 g \operatorname{sen}(\theta_1) + m_1 l_1 g \tag{6.82}$$

y la energía potencial del vínculo 2 como

$$u_2 = m_2 g d_2 \operatorname{sen}(\theta_1) + m_2 g d_{2m\acute{a}x}, \tag{6.83}$$

en donde $d_{2m\acute{a}x}$ es la extensión máxima de la articulación 2. Por ende, la energía potencial total se obtiene mediante

$$u(\Theta) = g(m_1 l_1 + m_2 d_2)\operatorname{sen}(\theta_1) + m_1 l_1 g + m_2 g d_{2m\acute{a}x}. \tag{6.84}$$

Ahora vamos a tomar derivadas parciales según sea necesario para la ecuación (6.77):

$$\frac{\partial k}{\partial \dot{\Theta}} = \begin{bmatrix} (m_1 l_1^2 + I_{zz1} + I_{zz2} + m_2 d_2^2)\dot{\theta}_1 \\ m_2 \dot{d}_2 \end{bmatrix}, \tag{6.85}$$

$$\frac{\partial k}{\partial \Theta} = \begin{bmatrix} 0 \\ m_2 d_2 \dot{\theta}_1^2 \end{bmatrix}, \tag{6.86}$$

$$\frac{\partial u}{\partial \Theta} = \begin{bmatrix} g(m_1 l_1 + m_2 d_2)\cos(\theta_1) \\ g m_2 \operatorname{sen}(\theta_1) \end{bmatrix}. \tag{6.87}$$

Por último, sustituyendo en la ecuación (6.77) tenemos que

$$\begin{aligned} \tau_1 &= (m_1 l_1^2 + I_{zz1} + I_{zz2} + m_2 d_2^2)\ddot{\theta}_1 + 2m_2 d_2 \dot{\theta}_1 \dot{d}_2 \\ &\quad + (m_1 l_1 + m_2 d_2) g \cos(\theta_1), \\ \tau_2 &= m_2 \ddot{d}_2 - m_2 d_2 \dot{\theta}_1^2 + m_2 g \operatorname{sen}(\theta_1). \end{aligned} \tag{6.88}$$

De la ecuación (6.89) podemos ver que

$$\begin{aligned} M(\Theta) &= \begin{bmatrix} (m_1 l_1^2 + I_{zz1} + I_{zz2} + m_2 d_2^2) & 0 \\ 0 & m_2 \end{bmatrix}, \\ V(\Theta, \dot{\Theta}) &= \begin{bmatrix} 2m_2 d_2 \dot{\theta}_1 \dot{d}_2 \\ -m_2 d_2 \dot{\theta}_1^2 \end{bmatrix}, \\ G(\Theta) &= \begin{bmatrix} (m_1 l_1 + m_2 d_2) g \cos(\theta_1) \\ m_2 g \operatorname{sen}(\theta_1) \end{bmatrix}. \end{aligned} \tag{6.89}$$

6.10 FORMULACIÓN DE LA DINÁMICA DE MANIPULADORES EN ESPACIO CARTESIANO

Nuestras ecuaciones dinámicas se han desarrollado en términos de las derivadas de posición y tiempo de los ángulos de articulación del manipulador, o en el **espacio de articulación**, con la siguiente forma general:

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta). \tag{6.90}$$

Desarrollamos esta ecuación en el espacio de articulación dado que podíamos usar la naturaleza de vínculo serial del mecanismo como ventaja al derivar las ecuaciones. En esta sección hablaremos sobre la formulación de las ecuaciones dinámicas que relacionan la aceleración del efector final expresada en espacio cartesiano, con las fuerzas y momentos cartesianos que actúan en el efector final.

La ecuación en el espacio de estados cartesiano

Como explicaremos en los capítulos 10 y 11, podría ser conveniente expresar la dinámica de un manipulador con respecto a variables cartesianas de la siguiente forma [6]:

$$\mathcal{F} = M_x(\Theta)\ddot{\chi} + V_x(\Theta, \dot{\Theta}) + G_x(\Theta), \tag{6.91}$$

en donde $\mathcal{F}$ es un vector de fuerza-momento de torsión que actúa en el efector final del robot, y χ es un vector cartesiano apropiado que representa la posición y la orientación del efector final [7]. $M_x(\Theta)$, que es análoga a la cantidades en el espacio de articulación, es la **matriz de masas cartesiana**, $V_x(\Theta, \dot{\Theta})$ es un vector de términos de velocidad en el espacio cartesiano, y $G_x(\Theta)$ es un vector de términos de gravedad en el espacio cartesiano. Tenga en cuenta que los actuadores podrían, de hecho, aplicar en las articulaciones las fuerzas ficticias $\mathcal{F}$ que actúan en el efector final, usando la siguiente relación:

$$\tau = J^T(\Theta)\mathcal{F}, \tag{6.92}$$

en donde el jacobiano $J(\Theta)$ está escrito en la misma trama que $\mathcal{F}$ y $\ddot{\chi}$, que generalmente es la trama herramienta $\{T\}$.

Podemos derivar la relación entre los términos de las ecuaciones (6.90) y (6.91) de la siguiente manera. Primero multiplicamos del lado izquierdo la ecuación (6.90) por la inversa de la transpuesta del jacobiano para obtener

$$J^{-T}\tau = J^{-T}M(\Theta)\ddot{\Theta} + J^{-T}V(\Theta, \dot{\Theta}) + J^{-T}G(\Theta), \tag{6.93}$$

o

$$\mathcal{F} = J^{-T}M(\Theta)\ddot{\Theta} + J^{-T}V(\Theta, \dot{\Theta}) + J^{-T}G(\Theta). \tag{6.94}$$

Después desarrollamos una relación entre el espacio de articulación y la aceleración cartesiana, empezando con la definición del jacobiano:

$$\dot{\chi} = J\dot{\Theta}, \tag{6.95}$$

y diferenciando para obtener

$$\ddot{\chi} = \dot{J}\dot{\Theta} + J\ddot{\Theta}. \tag{6.96}$$

Resolviendo la ecuación (6.96) para la aceleración del espacio de articulación llegamos a

$$\ddot{\Theta} = J^{-1}\ddot{\chi} - J^{-1}\dot{J}\dot{\Theta}. \tag{6.97}$$

Si sustituimos la ecuación (6.97) en la (6.94), tenemos

$$\mathcal{F} = J^{-T}M(\Theta)J^{-1}\ddot{\chi} - J^{-T}M(\Theta)J^{-1}\dot{J}\dot{\Theta} + J^{-T}V(\Theta, \dot{\Theta}) + J^{-T}G(\Theta), \tag{6.98}$$

de esta ecuación derivamos las expresiones para los términos en la dinámica cartesiana:

$$\begin{aligned} M_x(\Theta) &= J^{-T}(\Theta)M(\Theta)J^{-1}(\Theta), \\ V_x(\Theta, \dot{\Theta}) &= J^{-T}(\Theta)(V(\Theta, \dot{\Theta}) - M(\Theta)J^{-1}(\Theta)\dot{J}(\Theta)\dot{\Theta}), \\ G_x(\Theta) &= J^{-T}(\Theta)G(\Theta). \end{aligned} \tag{6.99}$$

Observe que el jacobiano que aparece en las ecuaciones (6.100) está escrito en las mismas tramas que $\mathcal{F}$ y χ en la ecuación (6.91); la elección de esta trama es arbitraria.[2] Observe que cuando el manipulador se acerca a una singularidad, ciertas cantidades en la dinámica cartesiana se vuelven infinitas.

[2] Ciertas elecciones podrían facilitar los cálculos computacionales.

EJEMPLO 6.6

Derive la forma de espacio cartesiano de la dinámica para el brazo planar de dos vínculos de la sección 6.7. Escriba la dinámica en términos de una trama unida al extremo del segundo vínculo.

Para este manipulador ya hemos obtenido la dinámica (en la sección 6.7) y el jacobiano [ecuación (5.66)], cuyas ecuaciones redefiniremos aquí:

$$J(\Theta) = \begin{bmatrix} l_1 s_2 & 0 \\ l_1 c_2 + l_2 & l_2 \end{bmatrix}. \tag{6.100}$$

Primero, calcule la inversa del jacobiano:

$$J^{-1}(\Theta) = \frac{1}{l_1 l_2 s_2} \begin{bmatrix} l_2 & 0 \\ -l_1 c_2 - l_2 & l_1 s_2 \end{bmatrix}. \tag{6.101}$$

Después obtenga la derivada con respecto al tiempo del jacobiano:

$$\dot{J}(\Theta) = \begin{bmatrix} l_1 c_2 \dot{\theta}_2 & 0 \\ -l_1 s_2 \theta_2 & 0 \end{bmatrix}. \tag{6.102}$$

Con la ecuación (6.100) y los resultados de la sección 6.7, obtenemos

$$M_x(\Theta) = \begin{bmatrix} m_2 + \frac{m_1}{s_2^2} & 0 \\ 0 & m_2 \end{bmatrix},$$

$$V_x(\Theta, \dot{\Theta}) = \begin{bmatrix} -(m_2 l_1 c_2 + m_2 l_2)\dot{\theta}_1^2 - m_2 l_2 \dot{\theta}_2^2 - (2 m_2 l_2 + m_2 l_1 c_2 + m_1 l_1 \frac{c_2}{s_2^2})\dot{\theta}_1 \dot{\theta}_2 \\ m_2 l_1 s_2 \dot{\theta}_1^2 + l_1 m_2 s_2 \dot{\theta}_1 \dot{\theta}_2 \end{bmatrix},$$

$$G_x(\Theta) = \begin{bmatrix} m_1 g \frac{c_1}{s_2} + m_2 g s_{12} \\ m_2 g c_{12} \end{bmatrix}. \tag{6.103}$$

Cuando $s_2 = 0$, el manipulador está en una posición singular, y algunos de los términos de dinámica se hacen infinitos. Por ejemplo, cuando $\theta_2 = 0$ (el brazo completamente estirado), la masa cartesiana efectiva del efector final se vuelve infinita en la dirección $\hat{X}_2$ de la trama de la punta con dos vínculos, según lo esperado. En general, en una configuración singular existe cierta dirección, la *dirección singular*, en la que es imposible moverse, pero es posible moverse en el subespacio "ortogonal" a esta dirección [8].

La ecuación de momento de torsión en el espacio de configuración cartesiana

Si combinamos las ecuaciones (6.91) y (6.92), podemos escribir momentos de torsión de articulación equivalentes con la dinámica expresada en espacio cartesiano:

$$\tau = J^T(\Theta)(M_x(\Theta)\ddot{\chi} + V_x(\Theta, \dot{\Theta}) + G_x(\Theta)). \tag{6.104}$$

Será de mucha utilidad escribir esta ecuación de la siguiente forma:

$$\tau = J^T(\Theta) M_x(\Theta)\ddot{\chi} + B_x(\Theta)[\dot{\Theta}\dot{\Theta}] + C_x(\Theta)[\dot{\Theta}^2] + G(\Theta), \tag{6.105}$$

en donde $B_x(\Theta)$ es una matriz de dimensión $n \times n(n-1)/2$ de coeficientes de Coriolis, $[\dot{\Theta}\dot{\Theta}]$ es un vector de $n(n-1)/2 \times 1$ de productos de velocidad de articulación definidos por

$$[\dot{\Theta}\dot{\Theta}] = [\dot{\theta}_1\dot{\theta}_2 \;\; \dot{\theta}_1 \, \dot{\theta}_3 \; \ldots \dot{\theta}_{n-1}\dot{\theta}_n]^T, \tag{6.106}$$

$C_x(\Theta)$ es una matriz de $n \times n$ de coeficientes centrífugos, y $[\dot{\Theta}^2]$ es un vector de $n \times 1$ definido por

$$[\dot{\theta}_1^2 \;\; \dot{\theta}_2^2 \;\; \ldots \dot{\theta}_n^2]^T. \tag{6.107}$$

Observe que en la ecuación (6.105), $G(\Theta)$ es igual que en la ecuación de espacio de articulación, pero en general, $B_x(\Theta) \neq B(\Theta)$ y $C_x(\Theta) \neq C(\Theta)$.

EJEMPLO 6.7

Encuentre $B_x(\Theta)$ y $C_x(\Theta)$ [de la ecuación (6.105)] para el manipulador de la sección 6.7.

Si formamos el producto $J^T(\Theta)V_x(\Theta, \dot{\Theta})$, encontraremos que

$$B_x(\Theta) = \begin{bmatrix} m_1 l_1^2 \frac{c_2}{s_2} - m_2 l_1 l_2 s_2 \\ m_2 l_1 l_2 s_2 \end{bmatrix} \tag{6.108}$$

y

$$C_x(\Theta) = \begin{bmatrix} 0 & -m_2 l_1 l_2 s_2 \\ m_2 l_1 l_2 s_2 & 0 \end{bmatrix}. \tag{6.109}$$

6.11 INCLUSIÓN DE LOS EFECTOS DE CUERPOS NO RÍGIDOS

Es importante tener en cuenta que las ecuaciones dinámicas que hemos derivado *no abarcan todos los efectos que actúan sobre un manipulador*, sólo incluyen aquellas fuerzas que surgen de mecanismos de cuerpo rígido y la fuente más importante de fuerzas que *no* se incluyen es la fricción. Es evidente que todos los mecanismos se ven afectados por fuerzas de fricción. En los manipuladores actuales, en los que es común tener una cantidad considerable de engranajes, las fuerzas debidas a la fricción pueden ser bastante grandes; tal vez sean equivalentes al 25% del momento de torsión requerido para mover el manipulador en situaciones comunes.

Para poder hacer que las ecuaciones dinámicas reflejen la realidad del dispositivo físico es importante modelar (cuando menos aproximadamente) estas fuerzas de fricción. Un modelo muy simple de la fricción es la **fricción viscosa**, en la que el momento de torsión debido a la fricción es proporcional a la velocidad del movimiento de la articulación. Por lo tanto, tenemos que

$$\tau_{fricción} = v\dot{\theta}, \tag{6.110}$$

en donde v es una constante de fricción viscosa. Otro posible modelo simple para la fricción es la **fricción de Coulomb**, que se utiliza ocasionalmente. La fricción de Coulomb es constante excepto por una dependencia de signo en la velocidad de una articulación, y se define así:

$$\tau_{fricción} = c \; sgn(\dot{\theta}), \tag{6.111}$$

en donde c es una constante de fricción de Coulomb. El valor de c se toma comúnmente a cierto valor cuando $\dot{\theta} = 0$, el coeficiente estático, pero a un menor valor, el coeficiente dinámico, cuando $\dot{\theta} \neq 0$. El que una articulación de un manipulador específico

exhiba fricción viscosa o de Coulomb es una cuestión complicada de lubricación y otros efectos. Un modelo razonable es incluir ambas, ya que las dos son muy probables:

$$\tau_{fricción} = c\, sgn(\dot{\theta}) + v\dot{\theta}. \tag{6.112}$$

En muchas articulaciones de manipuladores, la fricción también muestra una dependencia en la posición de la articulación. Una de las causas principales de este efecto podría ser que los engranajes no sean perfectamente redondos; su excentricidad haría que la fricción cambiara de acuerdo con la posición de la articulación. Entonces, un modelo de fricción relativamente complejo tendría la siguiente forma:

$$\tau_{fricción} = f(\theta, \dot{\theta}). \tag{6.113}$$

Estos modelos de fricción se añaden a los demás términos dinámicos derivados del modelo de cuerpo rígido, produciendo el siguiente modelo más completo:

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta) + F(\Theta, \dot{\Theta}). \tag{6.114}$$

Existen también otros efectos que se pasan por alto en este modelo. Por ejemplo, la suposición de vínculos de cuerpo rígido significa que no hemos incluido los efectos de flexión (que producen resonancias) en nuestras ecuaciones de movimiento. No obstante, estos efectos son extremadamente difíciles de modelar y están más allá del alcance de este libro (vea las referencias [9 y 10].)

6.12 SIMULACIÓN DINÁMICA

Para simular el movimiento de un manipulador, debemos hacer uso de un modelo de la dinámica tal como el que acabamos de desarrollar. Dada la dinámica escrita en forma cerrada como en la ecuación (6.59), la simulación requiere resolver la ecuación dinámica para la aceleración:

$$\ddot{\Theta} = M^{-1}(\Theta)[\tau - V(\Theta, \dot{\Theta}) - G(\Theta) - F(\Theta, \dot{\Theta})]. \tag{6.115}$$

Entonces podemos aplicar una de las diversas técnicas de **integración numérica** conocidas para integrar la aceleración y calcular las posiciones y velocidades futuras.

Dadas las condiciones iniciales en el movimiento del manipulador, generalmente en la forma

$$\begin{aligned} \Theta(0) &= \Theta_0, \\ \dot{\Theta}(0) &= 0, \end{aligned} \tag{6.116}$$

integramos la ecuación (6.115) avanzando en el tiempo numéricamente, en intervalos de tamaño Δt. Existen muchos métodos para realizar la integración numérica [11]. Aquí presentaremos el esquema de integración más simple, llamado *integración de Euler*. Empezando con $t = 0$, calcule iterativamente

$$\begin{aligned} \dot{\Theta}(t + \Delta t) &= \dot{\Theta}(t) + \ddot{\Theta}(t)\Delta t, \\ \Theta(t + \Delta t) &= \Theta(t) + \dot{\Theta}(t)\Delta t + \tfrac{1}{2}\ddot{\Theta}(t)\Delta t^2, \end{aligned} \tag{6.117}$$

en donde, para cada iteración, se emplea la ecuación (6.115) para calcular $\ddot{\Theta}$. De esta manera, la posición, la velocidad y la aceleración del manipulador, producidas por cierta función de momento de torsión de entrada, pueden calcularse numéricamente.

La integración de Euler es conceptualmente simple, pero para una simulación precisa y eficiente se recomiendan otras técnicas de integración más sofisticadas [11].

La forma de seleccionar el tamaño de Δt es una cuestión que se discute muy a menudo. Debe ser lo suficientemente pequeña para que el proceso de descomponer el tiempo continuo en estos pequeños incrementos sea una aproximación razonable y debe ser lo suficientemente grande como para no requerir una excesiva cantidad de tiempo de computadora para calcular una simulación.

6.13 CONSIDERACIONES COMPUTACIONALES

Como las ecuaciones dinámicas de movimiento para los manipuladores comunes son tan complejas, es importante considerar las cuestiones computacionales. En esta sección restringiremos nuestra atención a la dinámica de espacio de articulación. En las referencias [7 y 8] se habla sobre algunas cuestiones de eficiencia computacional de la dinámica cartesiana.

Una observación histórica en relación con la eficiencia

Si contamos el número de multiplicaciones y sumas para las ecuaciones (6.46) a (6.53) al tomar en cuenta el primer cómputo saliente simple y el último cómputo entrante simple, obtenemos:

$$126n - 99 \text{ multiplicaciones,}$$
$$106n - 92 \text{ sumas,}$$

en donde n es el número de vínculos (aquí son por lo menos dos). Aunque la formulación sigue siendo algo compleja, es muy eficiente en comparación con algunas formulaciones de dinámica de manipuladores sugeridas previamente. La primera formulación de la dinámica para un manipulador [12, 13] se hizo mediante un método Lagrangiano muy simple, cuyos cálculos requeridos resultaron ser aproximadamente [14]

$$32n^4 + 86n^3 + 171n^2 + 53n - 128 \text{ multiplicaciones,}$$
$$25n^4 + 66n^3 + 129n^2 + 42n - 96 \text{ sumas.}$$

Para un caso típico en el que $n = 6$, el esquema iterativo de Newton-Euler es aproximadamente 100 veces más eficiente. Por supuesto que los dos métodos deben producir ecuaciones equivalentes, y los cálculos numéricos producirían exactamente los mismos resultados, pero la estructura de las ecuaciones es muy distinta; esto no quiere decir que no pueda crearse un método lagrangiano para producir ecuaciones eficientes. Lo que esta comparación indica es que para formular un esquema computacional para este problema hay que tener mucho cuidado en cuanto a la eficiencia. La eficiencia relativa del método que hemos presentado proviene de plantear los cálculos como iteraciones de vínculo en vínculo y, en especial, de cómo se representan las diversas cantidades [15].

Renaud [16], Liegois y otros [17] hicieron las primeras contribuciones en lo que se refiere a la formulación de las descripciones de la distribución de masa de los vínculos. Mientras estudiaban el modelado de los miembros humanos, Stepanenko y Vukobratovic [18] empezaron a investigar un método de "Newton-Euler" para la dinámica, en vez del método lagrangiano que era algo más convencional. Orin y otros [19] revisaron la eficiencia de este trabajo aplicándolo a las piernas de robots que caminaban. El grupo de Orin mejoró un tanto la eficiencia al escribir las fuerzas y momentos en las tramas de referencia de los vínculos locales en vez de hacerlo en la trama inercial. También

observaron la naturaleza secuencial de los cálculos de un vínculo al siguiente y especularon que podría existir una eficiente formulación recursiva. Armstrong [20] y Luh, Walker y Paul [2] pusieron mucha atención a los detalles de la eficiencia y publicaron un algoritmo que es $O(n)$ en complejidad; para ello establecieron los cálculos en una naturaleza iterativa (o recursiva) y expresaron las velocidades y aceleraciones de los vínculos en las tramas de vínculos locales. Hollerbach [14] y Silver [15] exploraron más a fondo varios algoritmos computacionales. Hollerbach y Sahar [21] demostraron que para ciertas geometrías especializadas, la complejidad del algoritmo se reduciría aún más.

Comparación entre la eficiencia de forma cerrada y la de forma iterativa

El esquema iterativo que presentamos en este capítulo es bastante eficiente como un medio general para calcular la dinámica de cualquier manipulador, pero las ecuaciones de forma cerrada que se deriven para un manipulador específico serán por lo general más eficientes. Considere el manipulador planar de dos vínculos de la sección 6.7. Si utilizamos $n = 2$ en las fórmulas dadas en la sección 6.13, veremos que nuestro esquema iterativo requiere de 153 multiplicaciones y 120 sumas para calcular la dinámica de un manipulador general de dos vínculos. Sin embargo, ocurre que nuestro brazo específico de dos vínculos es bastante simple: es planar y las masas se tratan como masas puntuales. Por lo tanto, si consideramos las ecuaciones de forma cerrada que ideamos en la sección 6.7, veremos que el cálculo de la dinámica en esta forma requiere de aproximadamente 30 multiplicaciones y 13 sumas. Éste es un caso extremo ya que el manipulador específico es muy simple, pero ilustra la idea de que es muy probable que las ecuaciones simbólicas de forma cerrada sean la formulación más eficiente de la dinámica. Varios autores han publicado libros mostrando que, para cualquier manipulador dado, la dinámica de propósito específico con forma cerrada es más eficiente que incluso el mejor de los esquemas generales [22-27].

Por ende, si los manipuladores se diseñan para ser *simples*, en los sentidos cinemático y dinámico, tendrán ecuaciones dinámicas simples. Podríamos definir un **manipulador cinemáticamente** simple como un manipulador que tiene muchas de (o todas) sus torsiones de vínculo iguales a 0, 90 o $-90°$ y muchas de sus longitudes y desplazamientos de vínculo iguales a cero. Podríamos definir al manipulador **dinámicamente simple** como uno para el cual cada tensor de inercia de vínculo es diagonal en la trama $\{C_i\}$.

La desventaja de formular ecuaciones de forma cerrada es simplemente que en la actualidad se requiere una considerable cantidad de esfuerzo humano. No obstante, se han desarrollado programas de manipulación simbólica que pueden derivar las ecuaciones de forma cerrada del movimiento de un dispositivo y que automáticamente factorizan los términos comunes y realizan sustituciones trigonométricas [25, 28-30].

Dinámica eficiente para la simulación

Cuando se va a calcular la dinámica con el propósito de llevar a cabo la simulación numérica de un manipulador, nos interesa resolver para las aceleraciones de las articulaciones, dadas la posición y velocidad actuales del manipulador y los momentos de torsión de entrada. Por lo tanto, un esquema computacional eficiente deberá encargarse del cálculo de las ecuaciones dinámicas que estudiamos en este capítulo y de los esquemas eficientes para resolver ecuaciones (para las aceleraciones de las articulaciones), además de realizar la integración numérica. En la referencia [31] se reportan varios métodos eficientes para la simulación dinámica de manipuladores.

Esquemas de memorización

En cualquier esquema computacional puede hacerse un intercambio entre los cálculos y el uso de la memoria. En el problema de calcular la ecuación dinámica de un manipulador (6.59) hemos supuesto implícitamente que, cuando se necesita un valor de τ, se calcula lo más rápido posible a partir de Θ, $\dot{\Theta}$ y $\ddot{\Theta}$ en tiempo de ejecución. Si queremos, podemos intercambiar esta carga computacional por el costo de un requerimiento tremendamente grande de memoria, al calcular por anticipado la ecuación (6.59) para todos los valores posibles de Θ, $\ddot{\Theta}$ y $\ddot{\Theta}$ (cuantificados adecuadamente). Más adelante, cuando se necesite la información dinámica podremos encontrar la respuesta buscando en una tabla.

El tamaño de la memoria requerida es grande. Suponga que el rango de cada ángulo de articulación se cuantifica a diez valores discretos; de igual forma, suponga que las velocidades y las aceleraciones se cuantifican a diez rangos cada una. Para un manipulador de seis articulaciones, el número de celdas en el espacio cuantificado de (Θ, $\ddot{\Theta}$ y $\ddot{\Theta}$) es $(10 \times 10 \times 10)^6$. En cada una de estas celdas hay seis valores de momento de torsión. Suponiendo que cada valor de momento de torsión requiere de una palabra de computadora, ¡el tamaño de esta memoria sería de 6×10^{18} palabras! Además, observe que la tabla necesita recalcularse para un cambio en la masa de la carga; o se necesita agregar otra dimensión para tener en cuenta todas las cargas posibles.

Hay muchas soluciones intermedias que intercambian de varias formas la memoria por el cálculo. Por ejemplo, si las matrices que aparecen en la ecuación (6.63) se calcularan previamente, la tabla sólo tendría una dimensión (en Θ) y no tres. Después de localizar las funciones de Θ se lleva a cabo una modesta cantidad de cálculos [mediante la ecuación (6.63)]. Para obtener más detalles y para otras posibles parametrizaciones de este problema, vea las referencias [3] y [6].

BIBLIOGRAFÍA

[1] I. Shames, *Engineering Mechanics*, 2a. Edición, Prentice-Hall, Englewood Cliffs, NJ, 1967.

[2] J.Y.S. Luh, M.W. Walker y R.P. Paul, "On-Line Computational Scheme for Mechanical Manipulators", *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control*, 1980.

[3] M. Raibert, "Mechanical Arm Control Using a State Space Memory", artículo de *SME* MS77-750, 1977.

[4] K. R. Symon, *Mechanics*, 3a. Edición, Addison Wesley, Reading, MA, 1971.

[5] B.Noble, *Applied Linear Algebra*, Prentice-Hall, Englewood Cliffs, NJ, 1969.

[6] O. Khatib, "Commande Dynamique dans L'Espace Operationnel des Robots Manipulateurs en Presence d'Obstacles", Thèse de Docteur-Ingenieur. Ecole Nationale Superieure de l'Aeronautique et de L'Espace (ENSAE), Toulouse.

[7] O. Khatib, "Dynamic Control of Manipulators in Operational Space", Sexto Congreso de IFTOMM sobre la Teoría de Máquinas y Mecanismos, Nueva Delhi, diciembre 15-20, 1983.

[8] O. Khatib, "The Operational Space Formulation in Robot Manipulator Control", 15o. ISIR, Tokyo, septiembre 11-13, 1985.

[9] E. Schmitz, "Experiments on the End-Point Position Control of a Very Flexible One-Link Manipulator", Tesis Ph.D sin publicar, Departamento de Aeronáutica y Astronáutica, Stanford University, SUDAAR Núm. 547, junio de 1985.

[10] W. Book, "Recursive Lagrangian Dynamics of Flexible Manipulator Arms", *International Journal of Robotics Research*, Vol. 3, Núm. 3, 1984.

[11] S. Conte y C. DeBoor, *Elementary Numerical Analysis: An Algorithmic Approach*, 2da. Edición, McGraw-Hill, Nueva York, 1972.

[12] J. Uicker, "On the Dynamic Analysis of Spatial Linkages Using 4 x 4 Matrices", Disertación Ph.D sin publicar, Northwestern University, Evanston, IL, 1965.

[13] J. Uicker, "Dynamic Behaviour of Spatial Linkages", *ASME Mechanics*, Vol. 5, Núm. 68, págs. 1-15.

[14] J. M. Hollerbach, "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity", en *Robot Motion*, M. Brady y otros, Editores, MIT Press, Cambridge, MA, 1983.

[15] W. Silver, "On the Equivalence of Lagrangian and Newton-Euler Dynamics for Manipulators", *International Journal of Robotics Research*, Vol. 1, Núm. 2, págs. 60-70.

[16] M. Renaud, "Contribution à l'Etude de la Modélisation et de la Commande des Systèmes Mécaniques Articulés", Thèse de Docteur-Ingénieur, Université Paul Sabatier, Toulouse, diciembre de 1975.

[17] A. Liegois, W. Khalil, J.M. Dumas y M. Renaud, "Mathematical Models of Interconnected Mechanical Systems", Simposio sobre la Teoría y Práctica de Robots y Manipuladores, Polonia, 1976.

[18] Y. Stepanenko y M. Vukobratovic, "Dynamics of Articulated Open-Chain Active Mechanisms", *Math-Biosciences* Vol. 28, 1976, págs. 137-170.

[19] D.E. Orin y otros, "Kinematic and Kinetic Analysis of Open-Chain Linkages Utilizing Newton-Euler Methods", *Math-Biosciences* Vol. 43, 1979, págs. 107-130.

[20] W.W. Armstrong, "Recursive Solution to the Equations of Motion of an N-Link Manipulator", *Proceedings of the 5th World Congress on the Theory of Machines and Mechanisms*, Montreal, julio de 1979.

[21] J.M. Hollerbach y G. Sahar, "Wrist-Partitioned Inverse Accelerations and Manipulator Dynamics", MIT AI Memo Núm. 717, abril de 1983.

[22] T.K. Kanade, P.K. Koshla y N. Tanaka, "Real-Time Control of the CMU Direct Drive Arm II Using Customized Inverse Dynamics", *Proceedings of the 23rd IEEE Conference on Decision and Control*, Las Vegas, NV, diciembre de 1984.

[23] A. Izaguirre y R.P. Paul, "Computation of the Inertial and Gravitational Coefficients of the Dynamic Equations for a Robot Manipulator with a Load", *Proceedings of the 1985 International Conference on Robotics and Automation*, págs. 1024-1032, St. Louis, marzo de 1985.

[24] B. Armstrong, O. Khatib y J. Burdick, "The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm", *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, San Francisco, abril de 1986, págs. 510-518.

[25] J.W. Burdick, "An Algorithm for Generation of Efficient Manipulator Dynamic Equations", *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, San Francisco, abril 7-11, 1986, págs. 212-218.

[26] T.R. Kane y D.A. Levinson, "The Use of Kane's Dynamical Equations in Robotics", *The International Journal of Robotics Research*, Vol. 2, Núm. 3, otoño de 1983, págs. 3-20.

[27] M. Renaud, "An Efficient Iterative Analytical Procedure for Obtaining a Robot Manipulator Dynamic Model"; Primer Simposio Internacional de Investigación Robótica, NH, agosto de 1983.

[28] W. Schiehlen, "Computer Generation of Equations of Motion", en *Computer Aided Analysis and Optimization of Mechanical System Dynamics*, E.J. Haug, Editor, Springer-Verlag, Berlín y Nueva York, 1984.

[29] G. Cesareo, F. Nicolo y S. Nicosia, "DYMIR: A Code for Generating Dynamic Model of Robots", en *Advanced Software in Robotics*, Elsevier Science Publishers, Holanda del Norte, 1984.

[30] J. Murray y C. Neuman, "ARM: An Algebraic Robot Dynamic Modelling Program", *IEEE International Conference on Robotics*, Atlanta, marzo de 1984.

[31] M. Walker y D. Orin, "Efficient Dynamic Computer Simulation of Robotic Mechanisms", *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 104, 1982.

EJERCICIOS

6.1 [12] Encuentre el tensor de inercia de un cilindro recto de densidad homogénea respecto a una trama con el origen en el centro de masas del cuerpo.

6.2 [32] Construya las ecuaciones dinámicas para el manipulador de dos vínculos en la sección 6.7 cuando cada vínculo se modela como un sólido rectangular de densidad homogénea. Cada vínculo tiene las dimensiones l_i, ω_i y h_i, y una masa total representada por m_i.

6.3 [43] Construya las ecuaciones dinámicas para el manipulador de tres vínculos del capítulo 3, ejercicio 3.3. Considere que cada vínculo es un sólido rectangular de densidad homogénea con las dimensiones l_i, ω_i y h_i, y una masa total m_i.

6.4 [13] Escriba el conjunto de ecuaciones que corresponden a las ecuaciones (6.46) a la (6.53) para el caso en el que el mecanismo podría tener articulaciones deslizables.

6.5 [30] Construya las ecuaciones dinámicas para el manipulador no planar de dos vínculos que se muestra en la figura 6.8. Suponga que toda la masa de los vínculos puede considerarse como una masa puntual ubicada en el extremo distal (más externo) del vínculo. Los valores de las masas son m_1 y m_2, y las longitudes de los vínculos son l_1 y l_2. Este manipulador es como los primeros dos vínculos del brazo en el ejercicio 3.3. Suponga además que hay fricción viscosa actuando en cada vínculo, con los coeficientes v_1 y v_2.

6.6 [32] Derive la forma de espacio cartesiano de la dinámica para el manipulador planar de dos vínculos de la sección 6.7 en términos de la trama base. *Sugerencia*: vea el ejemplo 6.5, pero utilice el jacobiano escrito en la trama base.

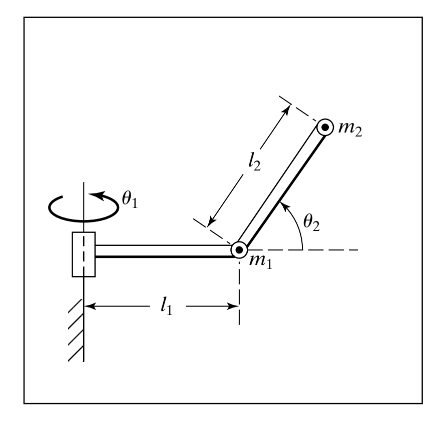


FIGURA 6.8: Manipulador no planar de dos vínculos con masas puntuales en los extremos distales de los vínculos.

6.7 [18] ¿Cuántas localidades de memoria se requerirían para almacenar las ecuaciones dinámicas de un manipulador general de tres vínculos en una tabla? Cuantifique la posición, velocidad y aceleración de cada articulación en 16 rangos. Haga todas las suposiciones necesarias.

6.8 [32] Derive las ecuaciones dinámicas para el manipulador de dos vínculos que se muestra en la figura 4.6. El vínculo 1 tiene un tensor de inercia dado por

$$C_1 I = \begin{bmatrix} I_{xx1} & 0 & 0 \\ 0 & I_{yy1} & 0 \\ 0 & 0 & I_{zz1} \end{bmatrix}.$$

Suponga que el vínculo 2 tiene toda su masa m_2 ubicada en un punto en el efector final. Suponga que la gravedad se dirige hacia abajo (en dirección opuesta a $\hat{Z}_1$).

6.9 [37] Derive las ecuaciones dinámicas para el manipulador de tres vínculos con una articulación prismática, mostrado en la figura 3.9. El vínculo 1 tiene un tensor de inercia dado por

$$C_1 I = \begin{bmatrix} I_{xx1} & 0 & 0 \\ 0 & I_{yy1} & 0 \\ 0 & 0 & I_{zz1} \end{bmatrix}.$$

El vínculo 2 tiene la masa puntual m_2 ubicada en el origen de su trama de vínculo. El vínculo 3 tiene un tensor de inercia dado por

$$C_3 I = \begin{bmatrix} I_{xx3} & 0 & 0 \\ 0 & I_{yy3} & 0 \\ 0 & 0 & I_{zz3} \end{bmatrix}.$$

Suponga que la gravedad se dirige en forma opuesta a $\hat{Z}_1$ y que hay fricción viscosa de magnitud v_i activa en cada articulación.

6.10 [35] Derive las ecuaciones dinámicas en espacio cartesiano para el manipulador del ejercicio 6.8. Escriba las ecuaciones en la trama {2}.

6.11 [20] Cierto manipulador de un vínculo tiene

$$C_1 I = \begin{bmatrix} I_{xx1} & 0 & 0 \\ 0 & I_{yy1} & 0 \\ 0 & 0 & I_{zz1} \end{bmatrix}.$$

Suponga que ésta es sólo la inercia del vínculo en sí. Si la armadura del motor tiene un momento de inercia I_m y la proporción de engranaje es de 100, ¿cuál es la inercia total vista desde el eje del motor [1]?

6.12 [20] El "manipulador" de un solo grado de libertad en la figura 6.9 tiene una masa total $m = 1$, con el centro de masas en

$$^1P_C = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix},$$

y tiene un tensor de inercia

$$^C I_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

Desde el resto a $t = 0$, el ángulo de articulación θ_1 se mueve de acuerdo con la función de tiempo

$$\theta_1(t) = bt + ct^2$$

en radianes. Proporcione la aceleración angular del vínculo y la aceleración lineal del centro de masas en términos de la trama {1} como una función de t.

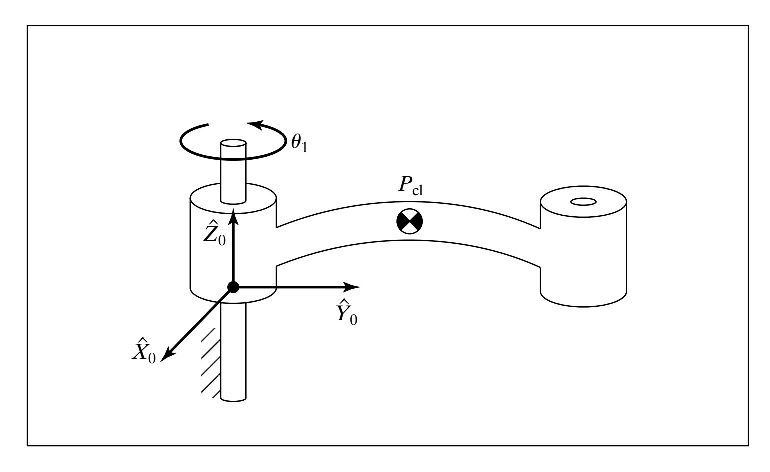


FIGURA 6.9: "Manipulador" de un vínculo del ejercicio 6.12.

6.13 [40] Construya las ecuaciones dinámicas cartesianas para el manipulador no planar de dos vínculos que se muestra en la figura 6.8. Suponga que toda la masa de los vínculos puede considerarse como una masa puntual ubicada en el extremo distal (más externo) del vínculo. Los valores de las masas son m_1 y m_2, y las longitudes de los vínculos son l_1 y l_2. Este manipulador es como los primeros dos vínculos del brazo en el ejercicio 3.3. Además suponga que hay fricción viscosa actuando en cada articulación con los coeficientes v_1 y v_2. Escriba la dinámica cartesiana en la trama {3}, que se ubica en la punta del manipulador y tiene la misma orientación que la trama de vínculo {2}.

6.14 [18] Las siguientes ecuaciones se derivaron para un manipulador 2-DOF RP:

$$\begin{aligned}\tau_1 &= m_1(d_1^2 + d_2)\ddot{\theta}_1 + m_2 d_2^2 \ddot{\theta}_1 + 2m_2 d_2 \dot{d}_2 \dot{\theta}_1 \\ &\quad + g\cos(\theta_1)[m_1(d_1 + d_2\dot{\theta}_1) + m_2(d_2 + \dot{d}_2)] \\ \tau_2 &= m_1 \dot{d}_2 \ddot{\theta}_1 + m_2 \ddot{d}_2 - m_1 d_1 \dot{d}_2 - m_2 d_2 \dot{\theta}^2 + m_2(d_2 + 1) g \operatorname{sen}(\theta_1).\end{aligned}$$

Algunos de los términos son obviamente incorrectos. Indique los términos incorrectos.

6.15 [28] Derive las ecuaciones dinámicas para el manipulador RP del ejemplo 6.5, usando el procedimiento de Newton-Euler en vez de la técnica lagrangiana.

6.16 [25] Derive las ecuaciones de movimiento para el manipulador PR que se muestra en la figura 6.10. Ignore la fricción pero incluya la gravedad. (Aquí, $\hat{X}_0$ va hacia arriba.) Los tensores de inercia de los vínculos son diagonales, con los momentos I_{xx1}, I_{yy1}, I_{zz1} y I_{xx2}, I_{yy2}, I_{zz2}. Los centros de la masa para los vínculos se proporcionan mediante

$$^1P_{C_1} = \begin{bmatrix} 0 \\ 0 \\ -l_1 \end{bmatrix},$$

$$^2P_{C_2} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

6.17 [40] Los términos relacionados con la velocidad que aparecen en la ecuación dinámica del manipulador pueden escribirse como un producto matriz-vector; esto es:

$$V(\Theta, \dot{\Theta}) = V_m(\Theta, \dot{\Theta})\dot{\Theta},$$

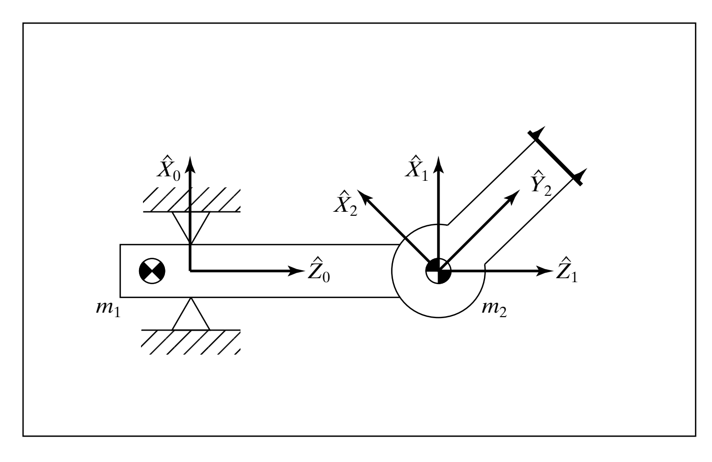


FIGURA 6.10: Manipulador PR del ejercicio 6.16.

en donde el subíndice m significa "forma matricial". Demuestre que existe una interesante relación entre la derivada en el tiempo de la matriz de masas del manipulador y $V_m(\cdot)$, a saber,

$$\dot{M}(\Theta) = 2V_m(\Theta, \dot{\Theta}) - S,$$

en donde S es una matriz antisimétrica.

6.18 [15] Proporcione dos propiedades que debería poseer cualquier modelo de fricción razonable (por ejemplo, el término $F(\Theta, \dot{\Theta})$ en la ecuación (6.114)).

6.19 [28] Haga el ejercicio 6.5, usando las ecuaciones de Lagrange.

6.20 [28] Derive las ecuaciones dinámicas del manipulador 2-DOF de la sección 6.7, usando una formulación lagrangiana.

EJERCICIO DE PROGRAMACIÓN (PARTE 6)

1. Derive las ecuaciones dinámicas de movimiento para el manipulador de tres vínculos (del ejemplo 3.3). Es decir, expanda la sección 6.7 para el caso de tres vínculos. Los siguientes valores numéricos describen al manipulador:

$$l_1 = l_2 = 0.5\text{m},$$
$$m_1 = 4.6\text{Kg},$$
$$m_2 = 2.3\text{Kg},$$
$$m_3 = 1.0\text{Kg},$$
$$g = 9.8\text{m/s}^2.$$

Para los primeros dos vínculos suponemos que toda la masa está concentrada en el extremo distal del vínculo. Para el vínculo 3 suponemos que el centro de masas está ubicado en el origen de la trama {3}; esto es, en el extremo próximo del vínculo. El tensor de inercia para el vínculo 3 es

$$^{C_s}I = \begin{bmatrix} 0.05 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \text{Kg-m}^2.$$

Los vectores que ubican cada centro de masas en relación con la trama de vínculo respectiva son

$$^{1}P_{C_1} = l_1\hat{X}_1,$$

$$^{2}P_{C_2} = l_2\hat{X}_2,$$

$$^{3}P_{C_3} = 0.$$

2. Escriba un simulador para el manipulador de tres vínculos. Basta con una simple rutina de integración de Euler para realizar la integración numérica (como en la sección 6.12). Para mantener el código modular, podría ser útil definir la rutina

```
Procedure ACTUALIZA(VAR tau: vec3; VAR periodo: real; VAR
theta, thetapunto: vec3);
```

en donde "tau" es el comando de momento de torsión para el manipulador (siempre cero para esta asignación), "periodo" es la longitud de tiempo que desea avanzar (en segundos) y "theta" y "thetapunto" son el *estado* del manipulador. Theta y thetapunto se actualizan cada "periodo" segundos, cada vez que usted llama a ACTUALIZA. Observe que el "periodo" generalmente debe ser más largo que el tamaño del intervalo de integración Δt utilizado en la integración numérica. Por ejemplo, aunque el tamaño de intervalo para la integración numérica podría ser de 0.001 segundos, tal vez sea conveniente imprimir la posición y velocidad del manipulador solamente cada 0.1 segundos.

Para probar su simulación, haga cero los comandos de momento de torsión de articulación (para todo el tiempo) y realice estas pruebas:

(a) Establezca la posición inicial del manipulador en

$$[\theta_1\ \theta_2\ \theta_3] = [-90\ 0\ 0].$$

Simule por unos cuantos segundos. ¿El movimiento del manipulador es lo que usted esperaría?

(b) Establezca la posición inicial del manipulador en

$$[\theta_1\ \theta_2\ \theta_3] = [30\ 30\ 10].$$

Simule por unos cuantos segundos. ¿El movimiento del manipulador es lo que usted esperaría?

(c) Introduzca algo de fricción viscosa en cada articulación del manipulador simulado; esto es, agregue un término a la dinámica de cada articulación en la forma $\tau_f = \nu\dot{\theta}$, en donde $\nu = 5.0$ newton-metros-segundos para cada articulación. Repita la prueba (b) anterior. ¿El movimiento es lo que usted esperaría?

EJERCICIO MATLAB 6A

Este ejercicio se enfoca en el análisis de dinámica inversa (en un contexto de control con velocidad resuelta; vea el ejercicio MATLAB 5) para el robot planar 2-DOF 2R. Este robot constituye las primeras dos articulaciones R y los primeros dos vínculos móviles del robot planar 3-DOF 3R. (Vea las figuras 3.6 y 3.7; los parámetros de DH se proporcionan en las primeras dos filas de la figura 3.8.)

Para el robot planar 2R, calcule los momentos de torsión de articulación requeridos (es decir, resuelva el problema de dinámica inversa) para proporcionar el movimiento comandado en cada intervalo, en un esquema de control con velocidad resuelta. Puede usar la recursividad numérica de Newton-Euler o las ecuaciones analíticas de los resultados del ejercicio 6.2, o ambas.

Dadas: $L_1 = 1.0$m, $L_2 = 0.5$ m; ambos vínculos son de acero sólido con una densidad de masa $\rho = 7806$ kg/m^3; ambos tienen las dimensiones de anchura y grosor $\omega = t = 5$ cm. Se supone que las articulaciones angulares son perfectas, conectando los vínculos precisamente en sus extremos (lo que no es físicamente posible).

Los ángulos iniciales son $\Theta = \begin{Bmatrix} \theta_1 \\ \theta_2 \end{Bmatrix} = \begin{Bmatrix} 10^\circ \\ 90^\circ \end{Bmatrix}$.

La velocidad cartesiana comandada (constante) es ${}^0\dot{X} = {}^0\begin{Bmatrix} \dot{x} \\ \dot{y} \end{Bmatrix} = {}^0\begin{Bmatrix} 0 \\ 0.5 \end{Bmatrix}$ (m/s).

Simule el movimiento por 1 segundo, con un intervalo de tiempo de control de 0.01 segundos. Presente cinco trazos (cada conjunto en un gráfico separado):

1. Los dos ángulos de articulación (grados) $\Theta = \{\theta_1\ \theta_2\}^T$ contra el tiempo.
2. Las dos velocidades de articulación (rad/s) $\dot{\Theta} = \{\dot{\theta}_1\ \dot{\theta}_2\}^T$ contra el tiempo.
3. Las dos aceleraciones de articulación (rad/s^2) $\ddot{\Theta} = \{\ddot{\theta}_1\ \ddot{\theta}_2\}^T$ contra el tiempo.
4. Los tres componentes cartesianos de ${}^0_HT$, $X = \{x\ y\ \phi\}^T$ (rad está bien para ϕ, de manera que se ajuste) contra el tiempo.
5. los dos momentos de torsión de articulación de dinámica inversa (Nm) $\mathrm{T} = \{\tau_1\ \tau_2\}^T$ contra el tiempo.

Etiquete cuidadosamente (a mano está bien) cada componente en cada trazo. Además, etiquete los nombres de los ejes y las unidades.

Realice la simulación dos veces. La primera vez ignore la gravedad (el plano de movimiento es normal al efecto de gravedad); la segunda vez considere la gravedad g en la dirección negativa de Y.

EJERCICIO MATLAB 6B

Este ejercicio se enfoca en la solución de dinámica inversa para el robot 3-DOF, 3R (de las figuras 3.6 y 3.7; los parámetros de DH se proporcionan en la figura 3.8), sólo para una instantánea de movimiento en el tiempo. Se proporcionan los siguientes parámetros de longitud fija: $L_1 = 4$, $L_2 = 3$ y $L_3 = 2$ (m). Para la dinámica también debemos recibir la información sobre la masa y el momento de inercia: $m_1 = 20$, $m_2 = 15$, $m_3 = 10$ (kg), ${}^CI_{ZZ1} = 0.5$, ${}^CI_{ZZ2} = 0.2$ y ${}^CI_{ZZ3} = 0.1$ (kgm^2). Suponga que el CG de cada vínculo está en su centro geométrico. Además suponga que la gravedad actúa en la dirección $-Y$ en el plano de movimiento. Para este ejercicio ignore la dinámica de los actuadores y el engranaje de las articulaciones.

a) Escriba un programa en MATLAB para implementar la solución de dinámica inversa recursiva de Newton-Euler (es decir, dado el movimiento comandado, calcule los momentos de torsión de articulación de control requeridos) para la siguiente instantánea de movimiento en el tiempo:

$$\Theta = \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{Bmatrix} = \begin{Bmatrix} 10^\circ \\ 20^\circ \\ 30^\circ \end{Bmatrix} \quad \dot{\Theta} = \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 2 \\ 3 \end{Bmatrix} \text{(rad/s)} \ \ddot{\Theta} = \begin{Bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{Bmatrix} = \begin{Bmatrix} 0.5 \\ 1 \\ 1.5 \end{Bmatrix} \text{(rad/s}^2\text{)}$$

b) Compruebe sus resultados en (*a*) mediante la Caja de Herramientas de Robótica MATLAB de Corke. Pruebe las funciones *rne()* y *gravload()*.

EJERCICIO MATLAB 6C

Este ejercicio se enfoca en la solución de dinámica directa para el robot planar 3-DOF, 3R (parámetros del ejercicio MATLAB 6B) para el movimiento sobre el tiempo. En este caso ignore la gravedad (es decir, suponga que la gravedad actúa en una dirección normal al plano

de movimiento). Use la Caja de Herramientas de Robótica MATLAB de Corke para resolver el problema de dinámica directa (es decir, dados los momentos de torsión de control comandados, calcule el movimiento resultante del robot) para los siguientes momentos de torsión de articulación constantes y los ángulos de articulación iniciales dados, y las proporciones de articulación iniciales dadas:

$$\mathrm{T} = \begin{Bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{Bmatrix} = \begin{Bmatrix} 20 \\ 5 \\ 1 \end{Bmatrix} \text{(Nm, constante)} \quad \Theta_0 = \begin{Bmatrix} \theta_{10} \\ \theta_{20} \\ \theta_{30} \end{Bmatrix} = \begin{Bmatrix} -60^\circ \\ 90^\circ \\ 30^\circ \end{Bmatrix}$$

$$\dot{\Theta}_0 = \begin{Bmatrix} \dot{\theta}_{10} \\ \dot{\theta}_{20} \\ \dot{\theta}_{30} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \text{ (rad/s)}$$

Realice esta simulación por 4 segundos. Pruebe con la función *fdyn()*.

Presente dos trazos para el movimiento resultante del robot (cada conjunto en un gráfico separado):

1. Los tres ángulos de articulación (grados) $\Theta = \{\theta_1 \;\; \theta_2 \;\; \theta_3\}^T$ contra el tiempo.
2. Las tres velocidades de articulación (rad/s) $\dot{\Theta} = \{\dot{\theta}_1 \;\; \dot{\theta}_2 \;\; \dot{\theta}_3\}^T$ contra el tiempo.

Etiquete cuidadosamente (a mano está bien) cada componente en cada trazo. Además, etiquete los nombres de los ejes y las unidades.

CAPÍTULO 7

Generación de trayectorias

7.1 INTRODUCCIÓN

En ese capítulo nos enfocaremos en los métodos para calcular una trayectoria que describa el movimiento deseado de un manipulador en espacio multidimensional. Aquí, **trayectoria** se refiere a un historial en el tiempo de la posición, la velocidad y la aceleración para cada grado de libertad.

Esto incluye el problema de interfaz humana de cómo deseamos *especificar* una trayectoria o ruta a través del espacio. Para facilitar la descripción del movimiento de un manipulador de un sistema de robot, no se le debería requerir al usuario que escriba funciones complicadas de espacio y tiempo para especificar la tarea. Lo que debemos hacer es proporcionar la capacidad de especificar trayectorias con simples descripciones del movimiento deseado y dejar que el sistema desarrolle los detalles. Por ejemplo, el usuario podría especificar nada más que la posición y la orientación de destino deseadas del efector final y dejar que el sistema decida la forma exacta de la ruta para llegar ahí, la duración, el perfil de velocidad y otros detalles.

También nos interesa la manera en que las trayectorias se *representan* en la computadora una vez planeadas. Finalmente, está el problema de calcular la trayectoria a partir de la representación interna (o *generar* la trayectoria). La generación ocurre en *tiempo de ejecución*; en el caso más general se calcula la posición, la velocidad y la aceleración. Estas trayectorias se determinan en computadoras digitales, por lo que los puntos de la trayectoria se calculan con cierta frecuencia, a la cual se le llama **frecuencia de actualización de ruta**. En sistemas manipuladores estándar esta frecuencia se encuentra entre 60 y 2000 Hz.

7.2 CONSIDERACIONES GENERALES SOBRE LA DESCRIPCIÓN Y GENERACIÓN DE RUTAS

En su mayor parte, consideraremos los movimientos de un manipulador como los movimientos de la trama herramienta $\{T\}$ relativos a la trama estación $\{S\}$, que es la

forma en la que pensaría un usuario eventual del sistema, por lo que diseñar un sistema de descripción y generación de rutas en estos términos produciría algunas ventajas importantes.

Cuando especificamos rutas como movimientos de la trama herramienta relativos a la trama estación, desacoplamos la descripción de movimiento de un robot, un efector final o unas piezas de trabajo específicas. Esto produce cierta modularidad y permite que se utilice la misma descripción de ruta con otro manipulador, o con el mismo, pero con un tamaño de herramienta distinto. Además, podemos especificar y planear los movimientos en relación con una estación de trabajo móvil (tal vez una banda transportadora) al planear los movimientos relativos a la trama estación como siempre y en tiempo de ejecución hacer que la definición de $\{S\}$ cambie con el tiempo.

Como se muestra en la figura 7.1, el problema básico es mover el manipulador desde una posición inicial hasta cierta posición final deseada; esto es, deseamos mover la trama herramienta desde su valor actual, $\{T_{\text{inicial}}\}$, hasta un valor final deseado, $\{T_{\text{final}}\}$. Observe que, en general, este movimiento implica tanto un cambio en la orientación como en la posición de la herramienta relativa a la estación.

Algunas veces es necesario especificar el movimiento con mucho más detalle que el necesario para sólo declarar la configuración final deseada. Una manera de incluir más detalle en una descripción de ruta es proporcionar una secuencia de **puntos vía** deseados (puntos intermedios entre las posiciones inicial y final). Por lo tanto, al completar el movimiento la trama herramienta debe pasar a través de un conjunto de posiciones y orientaciones intermedias, según lo descrito por los puntos vía. Cada uno de estos puntos vía es en realidad una trama que especifica tanto la posición como la orientación de la herramienta en forma relativa a la estación. El nombre **puntos ruta** incluye todos los puntos vía, junto con los puntos inicial y final. Recuerde que aunque utilizamos generalmente el término "puntos", estamos hablando en realidad de tramas que proporcionan la posición y la orientación. Junto con estas restricciones *espaciales* del movimiento, tal vez el usuario también desee especificar atributos *temporales* del movimiento. Por ejemplo, el tiempo transcurrido entre los puntos vía podría especificarse en la descripción de la ruta.

Generalmente es conveniente que el movimiento del manipulador sea *uniforme*. Aquí definiremos una función uniforme como una función que es continua y cuya primera derivada es continua. Algunas veces también es conveniente una segunda derivada

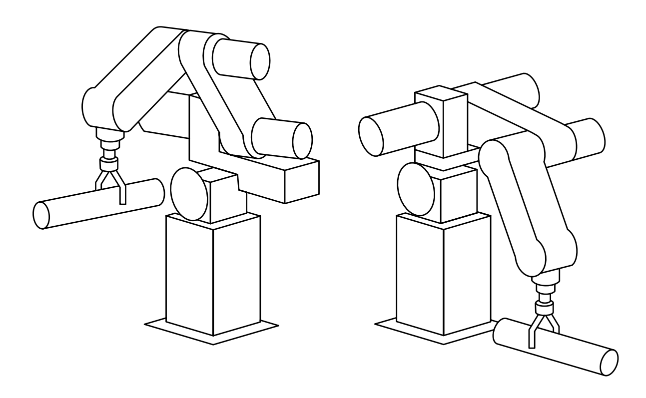


FIGURA 7.1: Al ejecutar una trayectoria, un manipulador se mueve desde su posición inicial hasta una posición de destino deseada de manera uniforme.

continua. Los movimientos bruscos y torpes tienden a producir un mayor desgaste en el mecanismo y producen vibraciones al excitar las resonancias en el manipulador. Para garantizar rutas uniformes primero debemos colocar cierto tipo de restricciones en las características espaciales y temporales de la ruta *entre* los puntos vía.

En este momento existen muchas elecciones que pueden tomarse y, por consecuencia, hay una gran variedad de formas en que pueden especificarse y planearse las rutas. Puede usar cualquier función uniforme de tiempo que pase a través de los puntos vía para especificar la forma exacta de la ruta. En este capítulo veremos un par de elecciones simples para estas funciones. Si desea conocer otros métodos consulte las referencias [1, 2] y [13-16].

7.3 ESQUEMAS EN EL ESPACIO DE ARTICULACIÓN

En esta sección consideraremos métodos de generación de rutas en los que las formas de éstas (en espacio y en tiempo) se describen en términos de funciones de ángulos de articulación.

Cada punto de la ruta se especifica generalmente en términos de una posición y una orientación deseadas de la trama herramienta $\{T\}$ relativa a la trama estación $\{S\}$. Cada uno de estos puntos vía se "convierte" en un conjunto de ángulos de articulación deseados mediante la aplicación de la cinemática inversa. Después se encuentra una función uniforme para cada una de las n articulaciones que pasan a través de los puntos vía y que terminan en el punto de destino. El tiempo requerido por segmento es el mismo para cada articulación, de manera que todas las articulaciones llegarán al punto vía al mismo tiempo, produciendo por consecuencia la posición cartesiana deseada de $\{T\}$ en cada punto vía. Aparte de especificar la misma duración para cada articulación, la determinación de la función de ángulo de articulación deseada para una articulación específica no depende de las funciones para las demás articulaciones.

Por ende, los esquemas de espacio de articulación logran la posición y la orientación deseadas en los puntos vía. En el trayecto entre los puntos vía, la forma de la ruta, aunque es considerablemente simple en el espacio de articulación, es compleja si se le describe en espacio cartesiano. Los esquemas en el espacio de articulación son generalmente los más fáciles de calcular y, como no hay una correspondencia continua entre el espacio de articulación y el espacio cartesiano, esencialmente no hay ningún problema con las singularidades del mecanismo.

Polinomios cúbicos

Considere el problema de mover la herramienta desde su posición inicial hasta una posición de destino en cierta cantidad de tiempo. La cinemática inversa permite calcular el conjunto de ángulos de articulación que corresponden a la posición y la orientación de destino. La posición inicial del manipulador también se conoce en la forma de un conjunto de ángulos de articulación. Lo que se requiere es una función para cada articulación, cuyo valor en t_0 sea la posición inicial de la articulación y cuyo valor en t_f sea la posición deseada de esa articulación. Como se muestra en la figura 7.2, hay muchas funciones uniformes $\theta(t)$ que podrían usarse para interpolar el valor de las articulaciones.

Para crear un solo movimiento uniforme son evidentes cuando menos cuatro restricciones sobre $\theta(t)$. Dos de esas restricciones sobre el valor de la función se deben a la selección de valores inicial y final:

$$\begin{aligned} \theta(0) &= \theta_0, \\ \theta(t_f) &= \theta_f. \end{aligned} \tag{7.1}$$

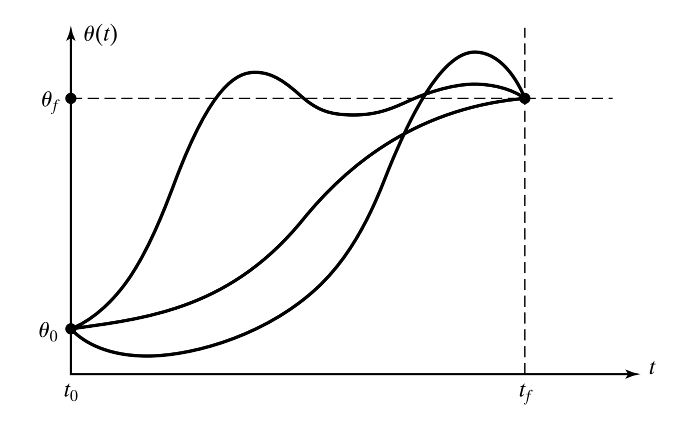


FIGURA 7.2: Varias formas posibles de ruta para una sola articulación.

Las otras dos restricciones son que la función debe ser continua en velocidad, lo cual en este caso significa que las velocidades inicial y final deben ser cero:

$$\begin{aligned}\dot{\theta}(0) &= 0,\\ \dot{\theta}(t_f) &= 0.\end{aligned} \tag{7.2}$$

Estas cuatro restricciones pueden satisfacerse mediante un polinomio de por lo menos tercer grado. (Un polinomio cúbico tiene cuatro coeficientes, por lo que puede usarse para satisfacer las cuatro restricciones dadas por las ecuaciones (7.1) y (7.2)). Estas restricciones determinan de manera única una ecuación cúbica específica. Este tipo de ecuaciones tiene la forma

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3, \tag{7.3}$$

de manera que la velocidad y la aceleración de la articulación a lo largo de esta ruta son claramente

$$\begin{aligned}\dot{\theta}(t) &= a_1 + 2a_2 t + 3a_3 t^2,\\ \ddot{\theta}(t) &= 2a_2 + 6a_3 t.\end{aligned} \tag{7.4}$$

Si combinamos las ecuaciones (7.3) y (7.4) con las cuatro restricciones deseadas se producen cuatro ecuaciones con cuatro incógnitas:

$$\begin{aligned}\theta_0 &= a_0,\\ \theta_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3,\\ 0 &= a_1,\\ 0 &= a_1 + 2a_2 t_f + 3a_3 t_f^2.\end{aligned} \tag{7.5}$$

Resolviendo estas ecuaciones para las a_i obtenemos

$$\begin{aligned}a_0 &= \theta_0,\\ a_1 &= 0,\end{aligned}$$

$$a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0), \tag{7.7}$$

$$a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0).$$

Con la ecuación (7.6) podemos calcular el polinomio cúbico que conecta cualquier posición de ángulo de articulación inicial con cualquier posición final deseada. Esta solución es para el caso en el que la articulación inicia y termina con una velocidad de cero.

EJEMPLO 7.1

Un robot de un solo vínculo con una articulación giratoria está sin movimiento a $\theta = 15$ grados. Se desea mover la articulación de manera uniforme a $\theta = 75$ grados en 3 segundos. Encuentre los coeficientes de una ecuación cúbica que realice este movimiento y lleve el manipulador a su destino. Trace la posición, la velocidad y la aceleración de la articulación como una función del tiempo.

Mediante la ecuación (7.6) encontramos que

$$\begin{aligned} a_0 &= 15.0, \\ a_1 &= 0.0, \\ a_2 &= 20.0, \\ a_3 &= -4.44. \end{aligned} \tag{7.7}$$

Con las ecuaciones (7.3) y (7.4) obtenemos

$$\begin{aligned} \theta(t) &= 15.0 + 20.0t^2 - 4.44t^3, \\ \dot{\theta}(t) &= 40.0t - 13.33t^2, \\ \ddot{\theta}(t) &= 40.0 - 26.66t. \end{aligned} \tag{7.8}$$

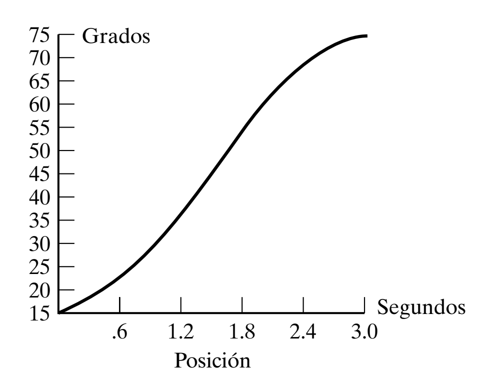
La figura 7.3 muestra las funciones de posición, velocidad y aceleración para este movimiento, muestreadas a 40 Hz. Observe que el perfil de velocidad para cualquier función es una parábola y que el perfil de aceleración es lineal.

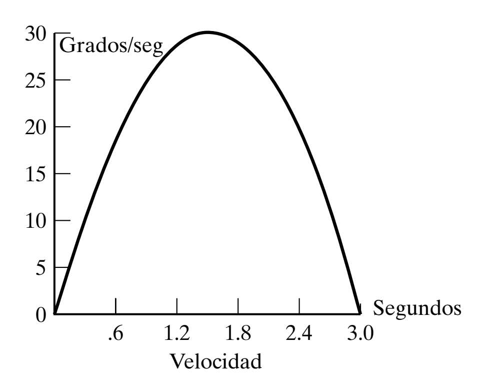
Polinomios cúbicos para una ruta con puntos vía

Hasta ahora hemos considerado los movimientos descritos por una duración deseada y un punto de destino final; en general deseamos permitir la especificación de rutas que incluyan puntos vía intermedios. Si el manipulador se va a detener en cada punto vía, podemos usar la solución cúbica de la sección 7.3.

Por lo común deseamos poder pasar a través de un punto vía sin detenernos, por lo cual necesitamos generalizar la forma en que ajustaremos ecuaciones cúbicas a las restricciones de las rutas.

Como en el caso de un solo punto de destino, cada punto vía se especifica generalmente en términos de una posición y una orientación deseadas de la trama herramienta relativa a la trama estación. Cada uno de estos puntos vía se “convierte” en un conjunto de ángulos de articulación deseados mediante la aplicación de cinemática inversa. Después consideramos el problema de calcular ecuaciones cúbicas que conecten entre sí los valores de los puntos vía para cada articulación de una manera uniforme.





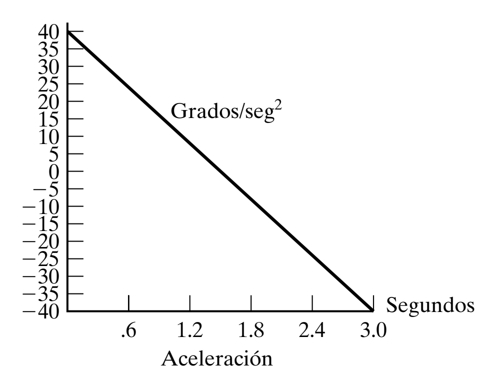


FIGURA 7.3: Perfiles de posición, velocidad y aceleración para un solo segmento cúbico que inicia y termina inmóvil (velocidad cero).

Si se conocen las velocidades deseadas de las articulaciones en los puntos vía, entonces podemos construir polinomios cúbicos como antes; sin embargo, ahora las restricciones de velocidad en cada extremo no son iguales a cero, sino que tienen cierta velocidad conocida. Las restricciones de la ecuación (7.3) se vuelven

$$\begin{aligned} \dot{\theta}(0) &= \dot{\theta}_0, \\ \dot{\theta}(t_f) &= \dot{\theta}_f. \end{aligned} \tag{7.9}$$

Las cuatro ecuaciones que describen este polinomio cúbico general son

$$\begin{aligned} \theta_0 &= a_0, \\ \theta_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3, \\ \dot{\theta}_0 &= a_1, \\ \dot{\theta}_f &= a_1 + 2a_2 t_f + 3a_3 t_f^2. \end{aligned} \tag{7.10}$$

Al resolver estas ecuaciones para las a_i obtenemos

$$\begin{aligned} a_0 &= \theta_0, \\ a_1 &= \dot{\theta}_0, \\ a_2 &= \frac{3}{t_f^2}(\theta_f - \theta_0) - \frac{2}{t_f}\dot{\theta}_0 - \frac{1}{t_f}\dot{\theta}_f, \\ a_3 &= -\frac{2}{t_f^3}(\theta_f - \theta_0) + \frac{1}{t_f^2}(\dot{\theta}_f + \dot{\theta}_0). \end{aligned} \tag{7.11}$$

Utilizando la ecuación (7.11) podemos calcular el polinomio cúbico que conecta las posiciones inicial y final con las velocidades inicial y final.

Si tenemos las velocidades de articulación deseadas en cada punto vía, entonces simplemente aplicamos la ecuación (7.11) a cada segmento para encontrar los polinomios cúbicos requeridos. Existen varias formas en las que podría especificarse la velocidad deseada en los puntos vía:

1. El usuario especifica la velocidad deseada en cada punto vía en términos de velocidades cartesianas lineal y angular de la trama herramienta en ese instante.
2. El sistema selecciona automáticamente las velocidades en los puntos vía mediante la aplicación de una heurística adecuada, ya sea en espacio cartesiano o de articulación.
3. El sistema selecciona de manera automática las velocidades en los puntos vía de tal forma que la aceleración en los puntos vía sea continua.

En la primera opción, las velocidades cartesianas deseadas en los puntos vía se “asignan” a las proporciones de articulación deseadas, por medio del jacobiano inverso del manipulador evaluado en el punto vía. Si el manipulador se encuentra en un punto singular en un punto vía específico, entonces el usuario no puede asignar una velocidad arbitraria en este punto. El esquema de generación de rutas tiene la útil capacidad de cumplir con una velocidad deseada que el usuario especifique, pero sería una carga requerir que el usuario siempre haga estas especificaciones. Por lo tanto, un sistema conveniente debería incluir la opción 2 o la 3 (o ambas).

En la opción 2 el sistema selecciona de manera automática velocidades intermedias razonables, por medio de algún tipo de heurística. Considere la ruta especificada por los puntos vía que se muestran para una articulación θ en la figura 7.4.

En esta figura hemos optado por la razonable elección de velocidades de articulación en los puntos vía, según lo indicado con pequeños segmentos lineales que representan tangentes a la curva en cada punto vía. Esta elección es el resultado de aplicar

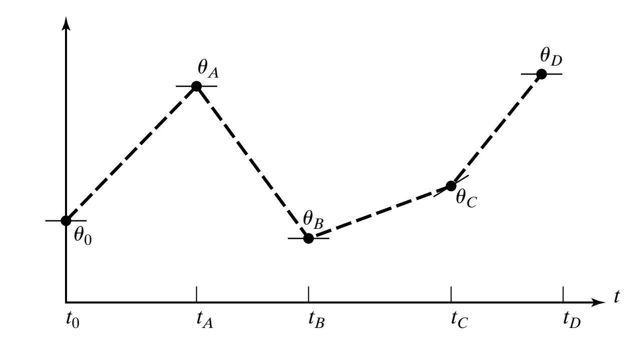


FIGURA 7.4: Puntos vía con velocidades deseadas en los puntos indicados por las tangentes.

una heurística simple en términos conceptuales y computacionales. Imagine los puntos vía conectados con segmentos de líneas rectas: si cambia de signo la pendiente de estas líneas en el punto vía, elija la velocidad cero; si la pendiente de estas líneas no cambia de signo, seleccione el promedio de las dos pendientes como velocidad del punto vía. De esta forma el sistema puede seleccionar la velocidad en cada punto con sólo especificar los puntos vía deseados.

En la opción 3 el sistema selecciona las velocidades de forma tal que la aceleración sea continua en el punto vía; para ello se necesita un nuevo enfoque. En este tipo de trazador o "*spline*"[1] sustituimos las dos restricciones de velocidad en la conexión de dos polinomios cúbicos con las dos restricciones que especifican que la velocidad y la aceleración deben ser continuas.

EJEMPLO 7.2

Resuelva para los coeficientes de dos polinomios cúbicos que están conectados en un trazador de dos segmentos con aceleración continua en el punto vía intermedio. El ángulo inicial es θ_0, el punto vía es θ_v, y el punto de destino es θ_g.

El primer polinomio cúbico es

$$\theta(t) = a_{10} + a_{11}t + a_{12}t^2 + a_{13}t^3, \tag{7.12}$$

y el segundo es

$$\theta(t) = a_{20} + a_{21}t + a_{22}t^2 + a_{23}t^3. \tag{7.13}$$

Cada polinomio cúbico se evaluará sobre un intervalo que empieza en $t = 0$ y termina en $t = t_{fi}$, en donde $i = 1$ o $i = 2$.

Las restricciones que queremos hacer valer son

$$\begin{aligned}
\theta_0 &= a_{10}, \\
\theta_\upsilon &= a_{10} + a_{11}t_{f1} + a_{12}t_{f1}^2 + a_{13}t_{f1}^3, \\
\theta_\upsilon &= a_{20}, \\
\theta_g &= a_{20} + a_{21}t_{f2} + a_{22}t_{f2}^2 + a_{23}t_{f2}^3, \qquad (7.14) \\
0 &= a_{11},
\end{aligned}$$

[1] Para nuestro uso, el término "trazador" (en inglés, *spline*) significa simplemente una función del tiempo.

la séptima igualdad de la ecuación 7.14 debe ser:

$$0 = a_{21} + 2a_{22}t_{f2} + 3a_{23}t_{f2}^2,$$
$$a_{11} + 2a_{12}t_{f1} + 3a_{13}t_{f1}^2 = a_{21},$$
$$2a_{12} + 6a_{13}t_{f1} = 2a_{22}.$$

Estas restricciones especifican un problema de ecuaciones lineales en el cual hay ocho ecuaciones y ocho variables desconocidas. Al resolver para el caso $t_f = t_{f1} = t_{f2}$, obtenemos

$$\begin{aligned}
a_{10} &= \theta_0, \\
a_{11} &= 0, \\
a_{12} &= \frac{12\theta_\upsilon - 3\theta_g - 9\theta_0}{4t_f^2}, \\
a_{13} &= \frac{-8\theta_\upsilon + 3\theta_g + 5\theta_0}{4t_f^3}, \\
a_{20} &= \theta_\upsilon, \\
a_{21} &= \frac{3\theta_g - 3\theta_0}{4t_f}, \\
a_{22} &= \frac{-12\theta_\upsilon + 6\theta_g + 6\theta_0}{4t_f^2}, \\
a_{23} &= \frac{8\theta_\upsilon - 5\theta_g - 3\theta_0}{4t_f^3}.
\end{aligned} \tag{7.15}$$

Para el caso general en el que se involucran n segmentos cúbicos, las ecuaciones que surgen por insistir en una aceleración continua en los puntos vía pueden formularse en forma matricial, la cual se resuelve para calcular las velocidades en los puntos vía. La matriz resulta ser tridiagonal y se resuelve fácilmente [4].

Polinomios de mayor orden

Algunas veces se utilizan polinomios de mayor orden para los segmentos de ruta. Por ejemplo, si deseamos especificar la posición, la velocidad *y* la aceleración al inicio y al final de cada segmento de ruta, se requiere un polinomio de quinto grado, a saber,

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5, \tag{7.16}$$

en donde las restricciones se dan así:

$$\theta_0 = a_0,$$
$$\theta_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5,$$

$$\begin{aligned}\dot{\theta}_0 &= a_1,\\ \dot{\theta}_f &= a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4,\\ \ddot{\theta}_0 &= 2a_2,\\ \ddot{\theta}_f &= 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3.\end{aligned} \tag{7.17}$$

Estas restricciones especifican un conjunto lineal de seis ecuaciones con seis variables desconocidas cuya solución es

$$\begin{aligned}a_0 &= \theta_0,\\ a_1 &= \dot{\theta}_0,\\ a_2 &= \frac{\ddot{\theta}_0}{2},\\ a_3 &= \frac{20\theta_f - 20\theta_0 - (8\dot{\theta}_f + 12\dot{\theta}_0)t_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^3},\\ a_4 &= \frac{30\theta_0 - 30\theta_f + (14\dot{\theta}_f + 16\dot{\theta}_0)t_f + (3\ddot{\theta}_0 - 2\ddot{\theta}_f)t_f^2}{2t_f^4},\\ a_5 &= \frac{12\theta_f - 12\theta_0 - (6\dot{\theta}_f + 6\dot{\theta}_0)t_f - (\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^5}.\end{aligned} \tag{7.18}$$

Existen varios algoritmos disponibles para calcular funciones uniformes (polinomios o de cualquier otro tipo) que pasen a través de un conjunto dado de puntos de datos [3, 4]. No obstante, este tema queda más allá del alcance de este libro.

Función lineal con mezclas parabólicas

Otra opción de forma de ruta es la lineal. Es decir, simplemente interpolamos linealmente para realizar un movimiento desde la posición de articulación actual hasta la posición final, como se muestra en la figura 7.5. Recuerde que aunque el movimiento de cada articulación en este esquema es lineal el efector final no se mueve en general en una línea recta en el espacio.

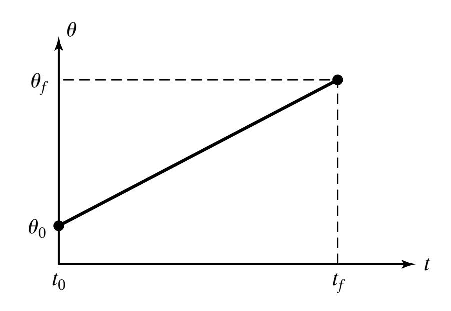


FIGURA 7.5: Interpolación lineal que requiere de una aceleración infinita.

Sin embargo, la interpolación lineal directa haría que la velocidad fuera discontinua al inicio y al final del movimiento. Para crear una ruta uniforme con posición y velocidad continuas, empezamos con la función lineal pero agregamos una región de *mezcla* parabólica en cada punto de la ruta.

Durante la porción de mezcla de la trayectoria se utiliza una aceleración constante para cambiar la velocidad de manera uniforme. La figura 7.6 muestra una ruta simple que se construye en esta forma. La función lineal y las dos funciones parabólicas se "empalman" de manera que toda la ruta completa sea continua en posición y en velocidad.

Para poder construir este segmento individual supongamos que las mezclas parabólicas tienen la misma duración; por lo tanto, se utiliza la misma aceleración constante durante ambas mezclas. Como se indica en la figura 7.7, hay muchas soluciones para este problema, pero debemos tener en cuenta que la respuesta es siempre simétrica sobre el punto intermedio en el tiempo, representado por t_h, y sobre el punto intermedio en la posición, representado por θ_h. La velocidad al final de la región de mezcla debe ser igual a la velocidad de la sección lineal, por lo que tenemos que

$$\ddot{\theta} t_b = \frac{\theta_h - \theta_b}{t_h - t_b}, \tag{7.19}$$

en donde θ_b es el valor de θ al final de la región de mezcla y $\ddot{\theta}$ es la aceleración que actúa durante la región de mezcla. El valor de θ_b se obtiene mediante

$$\theta_b = \theta_0 + \tfrac{1}{2}\ddot{\theta} t_b^2. \tag{7.20}$$

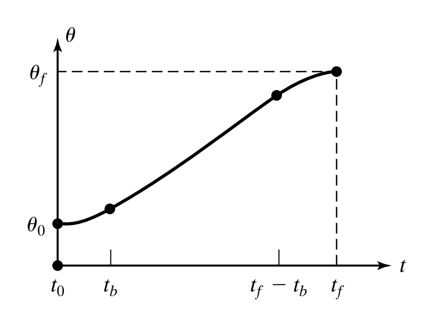


FIGURA 7.6: Segmento lineal con mezclas parabólicas.

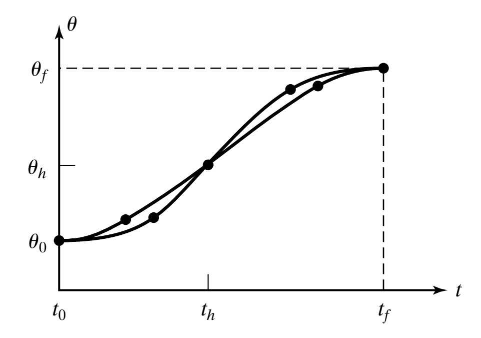


FIGURA 7.7: Segmento lineal con mezclas parabólicas.

Al combinar las ecuaciones (7.19) y (7.20) y hacer que $t = 2t_h$, obtenemos

$$\ddot{\theta} t_b^2 - \ddot{\theta} t t_b + (\theta_f - \theta_0) = 0, \tag{7.21}$$

en donde t es la duración deseada del movimiento. Dado cualquier valor para θ_f, θ_o y t, podemos seguir cualquiera de las rutas dadas con base en las elecciones de $\ddot{\theta}$ y t_b que satisfagan la ecuación (7.21). En general se elige una aceleración $\ddot{\theta}$ y se resuelve la ecuación (7.21) para obtener el valor correspondiente de t_b. La aceleración que se elija deberá ser lo suficiente alta o de lo contrario no existirá una solución. Al resolver la ecuación (7.21) para t_b en términos de la aceleración y otros parámetros conocidos, obtenemos

$$t_b = \frac{t}{2} - \frac{\sqrt{\ddot{\theta}^2 t^2 - 4\ddot{\theta}(\theta_f - \theta_0)}}{2\ddot{\theta}}. \tag{7.22}$$

La restricción sobre la aceleración utilizada en la mezcla es

$$\ddot{\theta} \geq \frac{4(\theta_f - \theta_0)}{t^2} \tag{7.23}$$

Cuando ocurre la igualdad en la ecuación (7.23) la porción lineal se reduce a una longitud cero y la ruta está compuesta de dos mezclas que se conectan con pendiente equivalente. A medida que la aceleración utilizada se hace cada vez más grande, la longitud de la región de mezcla se hace cada vez más corta. En el límite, con una aceleración infinita, regresamos al caso simple de interpolación lineal.

EJEMPLO 7.3

Para la misma ruta de un solo segmento descrita en el ejemplo 7.1, muestre dos ejemplos de una ruta lineal con mezclas parabólicas.

La figura 7.8(a) muestra una posibilidad en la que se eligió un valor de $\ddot{\theta}$ en extremo alto. En este caso aceleramos rápidamente, después seguimos a una velocidad constante y luego desaceleramos. La figura 7.8(b) muestra una trayectoria en la que la aceleración se mantiene a un valor considerablemente bajo, por lo que la sección lineal casi desaparece.

Función lineal con mezclas parabólicas para una ruta con puntos vía

Ahora consideremos las rutas lineales con mezclas parabólicas para el caso en el que haya un número arbitrario de puntos vía especificados. La figura 7.9 muestra un conjunto de puntos vía de espacio de articulación para cierta articulación θ. Las funciones lineales conectan los puntos vía y se agregan regiones de mezcla parabólica alrededor de éstos.

Utilizaremos la siguiente notación: considere tres puntos de ruta adyacentes, a los que llamaremos puntos j, k y l. La duración de la región de mezcla en el punto de ruta k es t_k. La duración de la porción lineal entre los puntos j y k es t_{jk}. La duración total del segmento que conecta los puntos j y k es t_{djk}. La velocidad durante la porción lineal es $\dot{\theta}_{jk}$ y la aceleración durante la mezcla en el punto j es $\ddot{\theta}_j$. Vea un ejemplo en la figura 7.9.

Al igual que con el caso de un solo segmento, existen muchas soluciones posibles, dependiendo del valor de aceleración usado en cada mezcla. Dados todos los puntos de ruta θ_k, las duraciones deseadas t_{djk} y la magnitud de la aceleración a usar en cada punto

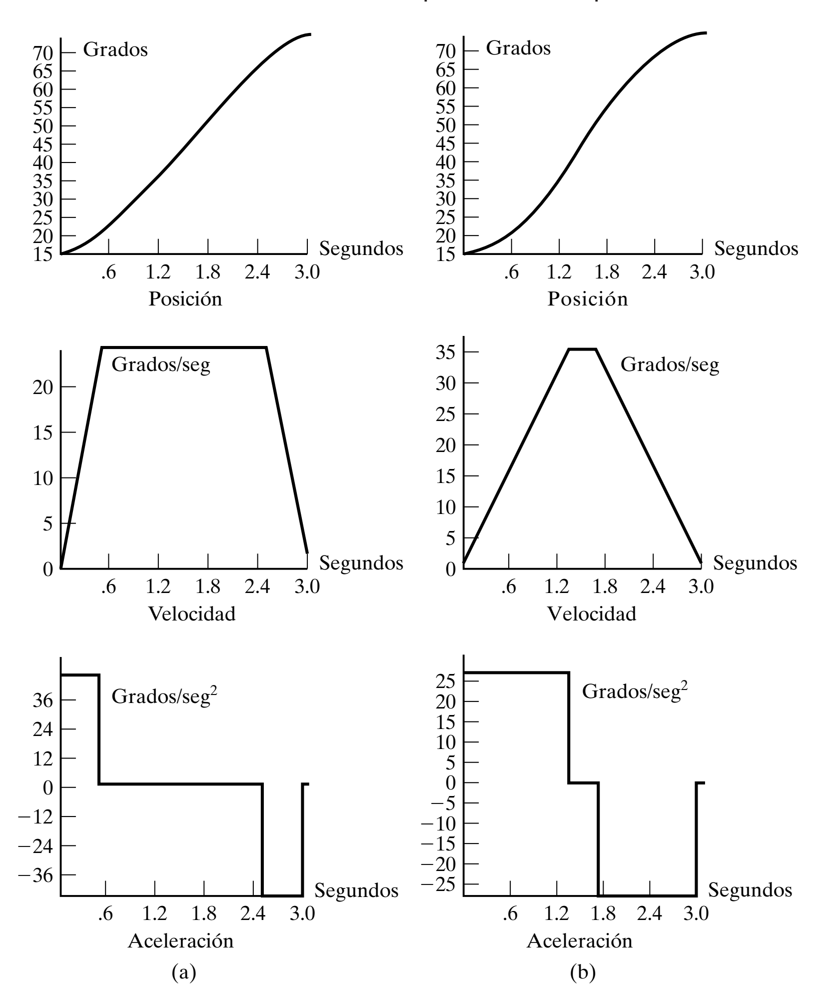


FIGURA 7.8: Perfiles de posición, velocidad y aceleración para la interpolación lineal con mezclas parabólicas. El conjunto de curvas de la izquierda está basado en una mayor aceleración durante las mezclas que la utilizada del lado derecho.

de ruta $|\ddot{\theta}_k|$, podemos calcular los tiempos de mezcla t_k. Para los puntos de la ruta interior, esto se obtiene fácilmente de las ecuaciones

$$\dot{\theta}_{jk} = \frac{\theta_k - \theta_j}{t_{djk}},$$

$$\ddot{\theta}_k = SGN(\dot{\theta}_{kl} - \dot{\theta}_{jk})|\ddot{\theta}_k|,$$

$$t_k = \frac{\dot{\theta}_{kl} - \dot{\theta}_{jk}}{\ddot{\theta}_k}, \tag{7.24}$$

$$t_{jk} = t_{djk} - \frac{1}{2}t_j - \frac{1}{2}t_k.$$

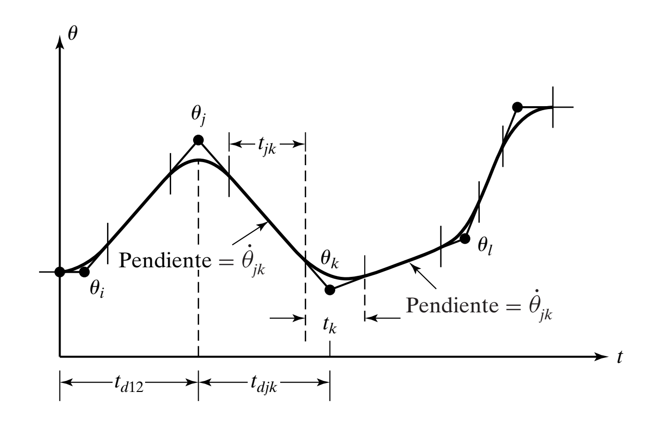


FIGURA 7.9: Ruta lineal de varios segmentos, con mezclas.

Los segmentos primero y último deben manejarse de manera ligeramente distinta, ya que debe contarse toda una región de mezcla en un extremo del segmento en la duración de tiempo total del segmento.

En el primer segmento resolvemos para t_1 igualando dos expresiones para la velocidad durante la fase lineal del segmento:

$$\frac{\theta_2 - \theta_1}{t_{12} - \frac{1}{2}t_1} = \ddot{\theta}_1 t_1. \tag{7.25}$$

Ésta puede resolverse para t_1, el tiempo de mezcla en el punto inicial; después $\dot{\theta}_{12}$ y t_{12} se calculan fácilmente:

$$\begin{aligned}
\ddot{\theta}_1 &= SGN(\theta_2 - \theta_1)|\ddot{\theta}_1|, \\
t_1 &= t_{d12} - \sqrt{t_{d12}^2 - \frac{2(\theta_2 - \theta_1)}{\ddot{\theta}_1}}, \\
\dot{\theta}_{12} &= \frac{\theta_2 - \theta_1}{t_{d12} - \frac{1}{2}t_1}, \\
t_{12} &= t_{d12} - t_1 - \frac{1}{2}t_2.
\end{aligned} \tag{7.26}$$

De igual forma, para el último segmento (el que conecta los puntos $n-1$ y n) tenemos que

$$\frac{\theta_{n-1} - \theta_n}{t_{d(n-1)n} - \frac{1}{2}t_n} = \ddot{\theta}_n t_n, \tag{7.27}$$

lo cual nos lleva a la solución

$$\begin{aligned}
\ddot{\theta}_n &= SGN(\theta_{n-1} - \theta_n)|\ddot{\theta}_n|, \\
t_n &= t_{d(n-1)n} - \sqrt{t_{d(n-1)n}^2 + \frac{2(\theta_n - \theta_{n-1})}{\ddot{\theta}_n}},
\end{aligned}$$

$$\dot{\theta}_{(n-1)n} = \frac{\theta_n - \theta_{n-1}}{t_{d(n-1)n} - \frac{1}{2}t_n}, \tag{7.28}$$

$$t_{(n-1)n} = t_{d(n-1)n} - t_n - \frac{1}{2}t_{n-1}.$$

Con las ecuaciones (7.24) a (7.28) podemos despejar los tiempos de mezcla y las velocidades para una ruta de varios segmentos. Por lo regular el usuario especifica sólo los puntos vía y la duración deseada de los segmentos. En este caso el sistema utiliza valores predeterminados para la aceleración de cada articulación. Algunas veces, para simplificar las cosas al usuario, el sistema calcula las duraciones con base en las velocidades predeterminadas. En todas las mezclas debe usarse la aceleración necesaria para que haya suficiente tiempo para entrar en la porción lineal del segmento antes de que empiece la siguiente región de mezcla.

EJEMPLO 7.4

La trayectoria de una articulación en particular se especifica de la siguiente manera: los puntos de ruta en grados: 10, 35, 25, 10. La duración de estos tres segmentos debe ser 2, 1, 3 segundos, respectivamente. La magnitud de la aceleración predeterminada que se va a utilizar en todos los puntos de mezcla es de 50 grados/seg^2. Calcule las velocidades, tiempos de mezcla y tiempos lineales de todos los segmentos.

Para el primer segmento aplicamos la ecuación (7.26) para encontrar

$$\ddot{\theta}_1 = 50.0. \tag{7.29}$$

Si aplicamos la ecuación (7.26) para calcular el tiempo de mezcla en el punto inicial, obtenemos

$$t_1 = 2 - \sqrt{4 - \frac{2(35 - 10)}{50.0}} = 0.27. \tag{7.30}$$

La velocidad $\dot{\theta}_{12}$ se calcula de la ecuación (7.26) así:

$$\dot{\theta}_{12} = \frac{35 - 10}{2 - 0.5(0.27)} = 13.50. \tag{7.31}$$

La velocidad $\dot{\theta}_{23}$ se calcula de la ecuación (7.24) así:

$$\dot{\theta}_{23} = \frac{25 - 35}{1} = -10.0. \tag{7.32}$$

A continuación aplicamos la ecuación (7.24) para encontrar

$$\ddot{\theta}_2 = -50.0. \tag{7.33}$$

Después, con la ecuación (7.24) se calcula t_2 y obtenemos

$$t_2 = \frac{-10.0 - 13.50}{-50.0} = 0.47. \tag{7.34}$$

La longitud de la porción lineal del segmento 1 se calcula de la ecuación (7.26):

$$t_{12} = 2 - 0.27 - \tfrac{1}{2}(0.47) = 1.50. \tag{7.35}$$

Ahora, de la ecuación (7.29) tenemos que

$$\ddot{\theta}_4 = 50.0. \tag{7.36}$$

Entonces, para el último segmento con la ecuación (7.28) calculamos t_4:

$$t_4 = 3 - \sqrt{9 + \frac{2(10-25)}{50.0}} = 0.102. \tag{7.37}$$

La velocidad $\dot{\theta}_{34}$ se calcula de la ecuación (7.28) así:

$$\dot{\theta}_{34} = \frac{10-25}{3-0.050} = -5.10. \tag{7.38}$$

A continuación se utiliza la ecuación (7.24) para obtener

$$\ddot{\theta}_3 = 50.0. \tag{7.39}$$

Entonces se calcula t_3 de la ecuación (7.24):

$$t_3 = \frac{-5.10 - (-10.0)}{50} = 0.098. \tag{7.40}$$

Por último, de la ecuación (7.24) calculamos

$$t_{23} = 1 - \tfrac{1}{2}(0.47) - \tfrac{1}{2}(0.098) = 0.716, \tag{7.41}$$

$$t_{34} = 3 - \tfrac{1}{2}(0.098) - 0.012 = 2.849. \tag{7.42}$$

Los resultados de estos cálculos constituyen un “plan” para la trayectoria. En tiempo de ejecución de **generador de rutas** utilizaría estos números para calcular los valores de θ, $\dot{\theta}$ y $\ddot{\theta}$ a la frecuencia de actualización de la ruta.

En estos trazadores de mezcla parabólica lineal hay que tener en cuenta que en realidad no se llega a los puntos vía, a menos que el manipulador se detenga. A menudo, cuando la capacidad de aceleración es lo suficiente alta, las rutas se acercarán considerablemente al punto vía deseado. Si en realidad queremos pasar a través de un punto, llegando a un alto, el punto vía simplemente se repite en la especificación de la ruta.

Si el usuario desea especificar que el manipulador pasa *exactamente* a través de un punto vía sin detenerse, puede acomodarse esta especificación con la misma formulación que antes, pero con la siguiente adición: el sistema sustituye automáticamente el punto vía a través del cual deseamos que pase el manipulador con dos *seudopuntos vía*: uno en cada lado del original (como en la figura 7.10). Entonces, la generación de rutas se lleva a cabo igual que antes. El punto vía original ahora se encontrará en la región lineal de la ruta que conecta los dos seudopuntos vía. Además de solicitar que el manipulador pase exactamente a través de un punto vía, el usuario puede solicitar también que pase con cierta velocidad. Si el usuario no especifica esta velocidad, el sistema la elegirá por medio de una heurística apropiada. Podría usarse el término **punto de paso** (en vez de punto vía) para especificar un punto de la ruta por el cual forzamos al manipulador para que pase exactamente *a través* suyo.

7.4 ESQUEMAS EN EL ESPACIO CARTESIANO

Como se mencionó en la sección 7.3, las rutas que se calculan en espacio de articulación pueden asegurar que se llegue a los puntos vía y los puntos de destino, incluso cuando

FIGURA 7.10: Uso de seudopuntos vía para crear un punto "de paso".

estos puntos de ruta se hayan especificado mediante tramas cartesianas. Sin embargo, la forma espacial de la ruta tomada por el efector final no es una línea recta a través del espacio, sino una forma complicada que depende de un comportamiento cinemático específico del manipulador que se esté utilizando. En esta sección consideraremos métodos de generación de rutas en los cuales las formas de las rutas se describen en términos de funciones que calculan la posición y la orientación cartesianas como funciones del tiempo; de esta manera podemos también especificar la forma espacial de la ruta entre los puntos de ruta. La forma más común de una ruta es una línea recta, pero podrían usarse rutas de forma circular, sinusoidal u otro tipo.

Cada punto de una ruta se especifica generalmente en términos de una posición y una orientación deseadas de la trama herramienta, relativas a la trama estación. En esquemas de generación de rutas con base cartesiana, las funciones que se "empalman" entre sí, para formar una trayectoria son funciones del tiempo que representan variables cartesianas. Estas rutas pueden ser *planeadas* directamente desde la definición de puntos de ruta del usuario, que son especificaciones de $\{T\}$ respecto a $\{S\}$, sin necesidad de realizar primero la cinemática inversa. No obstante, los esquemas cartesianos requieren de un mayor poder de cómputo para ejecutarse, ya que en tiempo de ejecución debe resolverse la cinemática inversa a la frecuencia de actualización de ruta; esto es, una vez que se genera la ruta en espacio cartesiano, como último paso debe realizarse el cálculo de la cinemática inversa para calcular los ángulos de articulación deseados.

La comunidad de robótica industrial e investigación ha propuesto varios esquemas para generar rutas cartesianas [1, 2]. En la siguiente sección presentaremos un esquema como ejemplo. En este esquema podremos usar el mismo trazador lineal-parabólico que desarrollamos para el caso del espacio de articulación.

Movimiento cartesiano en línea recta

A menudo es conveniente poder especificar con facilidad una ruta espacial que haga que la punta de la herramienta se mueva a través del espacio en línea recta. Obviamente, si especificamos muchos puntos vía con muy poca separación, distribuidos sobre una línea

recta, parecerá que la punta de la herramienta sigue una línea recta, sin importar la elección de función uniforme que interconecte los puntos vía. No obstante, es mucho más conveniente si la herramienta sigue rutas en línea recta entre puntos vía que tengan inclusive una amplia separación entre ellos. Este modo de especificación de ruta y ejecución se llama **movimiento cartesiano en línea recta**. La capacidad de definir movimientos en términos de líneas rectas es un subconjunto de la capacidad más general del **movimiento cartesiano**, en el cual pueden usarse funciones arbitrarias de variables cartesianas como funciones del tiempo para especificar una ruta. En un sistema que permita el movimiento cartesiano general podrían ejecutarse rutas con formas tales como elipses o sinusoides.

Al planear y generar rutas cartesianas en línea recta, es apropiado usar un trazador de funciones lineales con las mezclas parabólicas. Durante la porción lineal de cada segmento, los tres componentes de posición cambian en formal lineal, y el efector final se moverá a lo largo de una ruta lineal en el espacio. No obstante, si queremos especificar la orientación como una matriz de rotación en cada punto vía, no podemos interpolar linealmente sus elementos ya que esto no necesariamente produciría una matriz de rotación válida en todo momento. Una matriz de rotación debe estar compuesta de columnas ortonormales, y esta condición no podría garantizarse si se construyera mediante la interpolación lineal de elementos matriciales entre dos matrices válidas, por lo que utilizaremos otra representación de la orientación.

Según lo indicado en el capítulo 2, puede usarse la llamada representación **ángulo-eje** para especificar una orientación con tres números. Si combinamos esta representación de la orientación con la representación de la posición cartesiana de 3×1, tendremos una representación de 6×1 de la posición y orientación cartesianas. Considere un punto vía especificado en forma relativa a la trama estación como ${}^S_AT$. Es decir, la trama $\{A\}$ especifica un punto vía con la posición del efector final indicada por ${}^SP_{AORG}$, y la orientación del efector final indicada por ${}^S_AR$. Esta matriz de rotación puede convertirse a la representación de ángulo-eje $ROT({}^S\hat{K}_A, \theta_{SA})$; o simplemente ${}^SK_A$. Utilizaremos el símbolo χ para representar este vector de 6×1 de posición y orientación cartesianas. En consecuencia tenemos que

$$ {}^S\chi_A = \begin{bmatrix} {}^SP_{AORG} \\ {}^SK_A \end{bmatrix}, \tag{7.43} $$

en donde ${}^SK_A$ está formada por la aplicación de un cambio de escala al vector unitario ${}^S\hat{K}_A$ en una proporción igual a la cantidad de rotación θ_{SA}. Si cada punto de la ruta se especifica en esta representación, entonces hay que describir funciones de trazador que varíen uniformemente estas seis cantidades de un punto de ruta a otro, como funciones del tiempo. Si se utilizan trazadores lineales con mezclas parabólicas, la forma de la ruta entre los puntos vía será lineal. Al pasar por los puntos vía, la velocidad lineal y la velocidad angular del efector final cambian uniformemente.

Tenga en cuenta que, a diferencia de otros esquemas de movimiento cartesiano en línea recta que se han propuesto, este método no garantiza que las rotaciones ocurran sobre un solo "eje equivalente" al moverse de un punto a otro. Nuestro esquema es uno simple que proporciona cambios de orientación uniformes y permite el uso de los mismos cálculos que ya hemos desarrollado para planear trayectorias interpoladas de articulaciones.

Existe una ligera complicación por el hecho de que la representación ángulo-eje de la orientación no es única; esto es,

$$ ({}^S\hat{K}_A, \theta_{SA}) = ({}^S\hat{K}_A, \theta_{SA} + n360°), \tag{7.44} $$

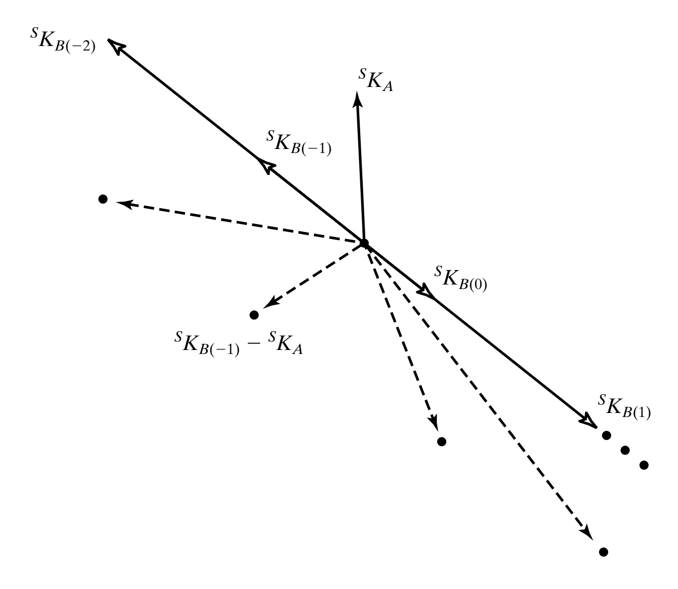


FIGURA 7.11: Selección de la representación ángulo-eje para minimizar la rotación.

en donde n es cualquier entero positivo o negativo. Al pasar de un punto vía $\{A\}$ a un punto vía $\{B\}$, debe minimizarse la cantidad total de rotación. Si nuestra representación de la orientación de $\{A\}$ se da como ^SK_A, debemos seleccionar la ^SK_B específica, de tal forma que $|^SK_B - {}^SK_A|$ se minimice. Por ejemplo, la figura 7.11 muestra cuatro posibles valores de ^SK_B y su relación con la ^SK_A dada. Los vectores de diferencia (líneas punteadas) se comparan para saber cuál ^SK_B producirá la mínima rotación; en este caso, $^SK_{B(-1)}$.

Una vez que seleccionemos los seis valores de χ para cada punto vía, podremos usar los mismos cálculos que ya hemos desarrollado para generar trazadores que estén compuestas de secciones lineales y parabólicas. No obstante, debemos agregar una restricción más: los tiempos de mezcla para cada grado de libertad deben ser iguales. Esto asegurará que el movimiento resultante de todos los grados de libertad sea una línea recta en el espacio. Como todos los tiempos de mezcla deben ser iguales, la aceleración utilizada durante la mezcla para cada grado de libertad será distinta. Por lo tanto especificamos una duración de mezcla y, con la ecuación (7.24), calculamos la aceleración necesaria (en lugar de hacerlo al revés). El tiempo de mezcla debe elegirse de manera que no se exceda de un cierto límite superior en la aceleración.

Pueden usarse muchos otros esquemas para representar e interpolar la porción de orientación de una ruta cartesiana; por ejemplo, podrían usarse algunas de las otras representaciones de 3×1 de la orientación que se presentaron en la sección 2.8. Por ejemplo, algunos robots industriales se mueven a lo largo de rutas cartesianas en línea recta en las que la interpolación de la orientación se realiza mediante una representación similar a los ángulos de Euler Z-Y-Z.

7.5 PROBLEMAS GEOMÉTRICOS CON LAS RUTAS CARTESIANAS

Dado que se crea una correspondencia continua entre una forma de ruta descrita en espacio cartesiano y las posiciones de articulación, las rutas cartesianas son propensas a varios problemas relacionados con el espacio de trabajo y las singularidades.

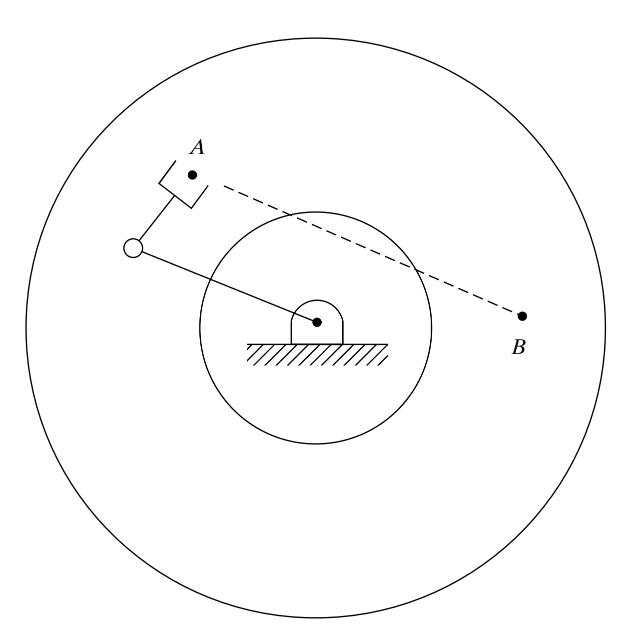


FIGURA 7.12: Problema de ruta cartesiana de tipo 1.

Problemas de tipo 1: puntos intermedios inalcanzables

Aunque la ubicación inicial de un manipulador y el punto de destino final se encuentren dentro del espacio de trabajo del manipulador, es muy posible que no todos los puntos distribuidos en una línea recta que conecten estos dos puntos se encuentren en el espacio de trabajo. Como ejemplo, considere el robot planar de dos vínculos que se muestra en la figura 7.12 y su espacio de trabajo asociado. En este caso el vínculo 2 es más corto que el vínculo 1, por lo que el espacio de trabajo contiene un agujero en la parte central, cuyo radio es la diferencia entre las longitudes de los vínculos. En el espacio de trabajo están dibujados un punto inicial A y un punto de destino B. Moverse de A a B en espacio de articulación no sería problema pero, si se intentara un movimiento en línea recta cartesiana, los puntos intermedios a lo largo de la ruta serían inalcanzables. Éste es un ejemplo de una situación en la que una ruta en espacio de articulación puede ejecutarse fácilmente, pero una ruta en línea recta cartesiana fallaría.[2]

Problemas de tipo 2: velocidades de articulación altas cerca de una singularidad

En el capítulo 5 vimos que existen ubicaciones en el espacio de trabajo del manipulador en donde es imposible elegir velocidades de articulación finitas que produzcan la velocidad deseada del efector final en espacio cartesiano. Por lo tanto, no debe sorprender que haya ciertas rutas (descritas en términos cartesianos) que el manipulador no pueda recorrer. Por ejemplo, si un manipulador está siguiendo una ruta en línea recta cartesiana y se acerca a una configuración singular del mecanismo, una o más velocidades de articulación podrían incrementarse hasta el infinito. Como las velocidades

[2] Algunos sistemas de robots notificarían al usuario sobre un problema antes de mover el manipulador; en otros, el movimiento empezaría a lo largo de la ruta hasta que alguna articulación llegara a su límite; en ese instante se detendría el movimiento del manipulador.

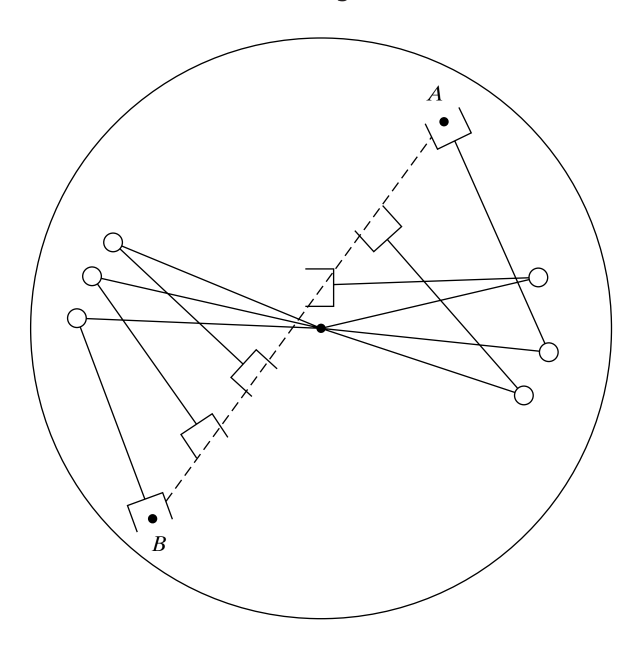


FIGURA 7.13: Problema de ruta cartesiana de tipo 2.

del mecanismo tienen límites superiores, esta situación generalmente provoca que el manipulador se desvíe de la ruta deseada.

Como un ejemplo, la figura 7.13 muestra un robot planar de dos vínculos (con longitudes de vínculo iguales) que se mueve a lo largo de una ruta, del punto A al punto B. La trayectoria deseada es mover la punta del extremo del manipulador, con una velocidad lineal constante, a lo largo de la ruta en línea recta. En la figura se han dibujado varias posiciones intermedias del manipulador para ayudarnos a visualizar su movimiento. Todos los puntos a lo largo de la ruta son alcanzables, pero a medida que el robot avanza más allá de la porción media de la ruta, la velocidad de la articulación uno es más alta. Entre más se acerque la ruta al eje de la articulación uno, más rápido será este movimiento. Una solución es reducir la velocidad total de la ruta hasta que todas las articulaciones permanezcan dentro de sus capacidades de velocidad. De esta manera podrían perderse los atributos temporales deseados de la ruta, pero al menos se adheriría al aspecto espacial de la definición de la trayectoria.

Problemas de tipo 3: inicio y destino alcanzables en distintas soluciones

En la figura 7.14 se muestra un tercer tipo de problema que podría surgir. Aquí, un robot planar de dos vínculos con iguales longitudes de vínculos tiene límites de articulación que restringen el número de soluciones con las que puede llegar a un punto dado en el espacio. Específicamente, surgiría un problema si el punto de destino no puede alcanzarse en la misma solución física en la que se encuentra el robot en el punto de inicio. En la figura 7.14, el manipulador puede llegar a todos los puntos de la ruta en cierta solución, pero no en una sola solución. En esta situación el sistema de planeación de trayectorias del manipulador puede detectar este problema sin necesidad de tratar de mover el robot a lo largo de la ruta, y puede indicar un error al usuario.

Para manejar estos problemas con las rutas especificadas en espacio cartesiano, la mayoría de los sistemas de control de manipulador industriales soportan la generación

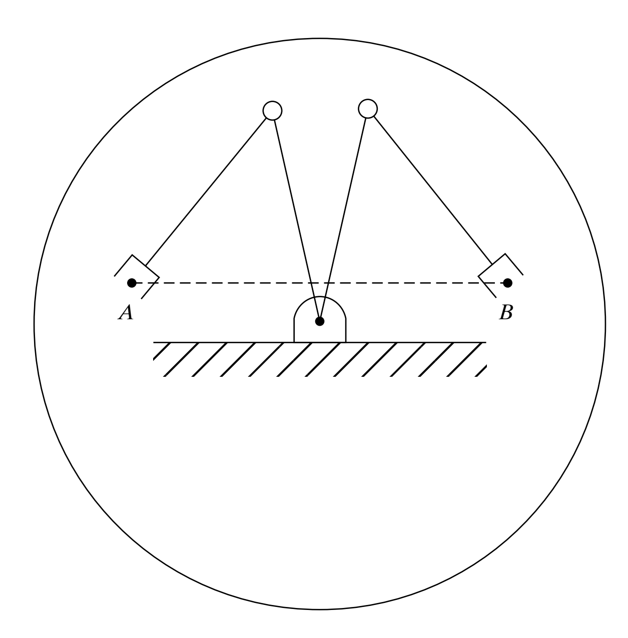


FIGURA 7.14: Problema de ruta cartesiana de tipo 3.

de rutas en espacio de articulación y también en espacio cartesiano. El usuario aprende rápidamente que, debido a las dificultades con las rutas cartesianas, debe usar rutas predeterminadas en espacio de articulación, y las rutas en espacio cartesiano deben usarse sólo cuando sea realmente necesario para la aplicación.

7.6 GENERACIÓN DE RUTAS EN TIEMPO DE EJECUCIÓN

En **tiempo de ejecución**, la rutina generadora de rutas construye la trayectoria, generalmente en términos de θ, $\dot{\theta}$ y $\ddot{\theta}$, y alimenta esta información al sistema de control del manipulador. Este generador de rutas calcula la trayectoria a la frecuencia de actualización de rutas.

Generación de rutas en espacio de articulación

El resultado de haber planeado una ruta mediante el uso de cualquiera de los métodos de interpolación por trazadores mencionados en la sección 7.3, es un conjunto de datos para cada segmento de la trayectoria. El generador de rutas utiliza estos datos en tiempo de ejecución para calcular θ, $\dot{\theta}$ y $\ddot{\theta}$.

En el caso de trazadores cúbicos, el generador de rutas simplemente calcula la ecuación (7.3) conforme se incrementa *t*. Cuando se llega al final de un segmento se solicita un nuevo conjunto de coeficientes cúbicos, *t* regresa a cero y la generación continúa.

En el caso de trazadores lineales con mezclas parabólicas, se comprueba el valor del tiempo *t* en cada actualización para determinar si nos encontramos actualmente en la porción lineal o en la porción de mezcla del segmento. En la porción lineal, la trayectoria para cada unión se calcula así:

$$\begin{aligned} \theta &= \theta_j + \dot{\theta}_{jk} t, \\ \dot{\theta} &= \dot{\theta}_{jk}, \\ \ddot{\theta} &= 0, \end{aligned} \qquad (7.45)$$

en donde t es el tiempo transcurrido desde el j-ésimo punto vía y $\dot{\theta}_{jk}$ se calculó en el tiempo de planeación de la ruta mediante la ecuación (7.24). En la región de mezcla, la trayectoria para cada articulación se calcula así:

$$\begin{aligned} t_{inb} &= t - (\tfrac{1}{2}t_j + t_{jk}), \\ \theta &= \theta_j + \dot{\theta}_{jk}(t - t_{inb}) + \tfrac{1}{2}\ddot{\theta}_k t_{inb}^2, \\ \dot{\theta} &= \dot{\theta}_{jk} + \ddot{\theta}_k t_{inb}, \\ \ddot{\theta} &= \ddot{\theta}_k, \end{aligned} \tag{7.46}$$

en donde $\dot{\theta}_{jk}$, $\ddot{\theta}_k$ y t_{jk} se calcularon en el tiempo de planeación de la ruta mediante las ecuaciones (7.24) a (7.28). Esto continúa, y t se restablece a $\frac{1}{2}t_k$ cuando se introduce un nuevo segmento lineal, hasta que hayamos pasado a través de todos los conjuntos de datos que representan los segmentos de la ruta.

Generación de rutas de espacio cartesiano

Para el esquema de ruta cartesiana presentado en la sección 7.4, utilizamos el generador de rutas para la ruta del trazador lineal con mezclas parabólicas. No obstante, los valores calculados representan la posición y orientación cartesianas en vez de los valores de variable de articulación, por lo que reescribimos las ecuaciones (7.45) y (7.46) con el símbolo x que representa un componente del vector de posición y orientación cartesiana. En la porción lineal del segmento, cada grado de libertad en χ se calcula así:

$$\begin{aligned} x &= x_j + \dot{x}_{jk}t, \\ \dot{x} &= \dot{x}_{jk}, \\ \ddot{x} &= 0, \end{aligned} \tag{7.47}$$

en donde t es el tiempo transcurrido desde el j-ésimo punto vía y $\dot{x}_{jk}$ se calculó en el tiempo de planeación de la ruta mediante una ecuación análoga a la (7.24). En la región de mezcla, la trayectoria para cada grado de libertad se calcula así:

$$\begin{aligned} t_{inb} &= t - (\tfrac{1}{2}t_j + t_{jk}), \\ x &= x_j + \dot{x}_{jk}(t - t_{inb}) + \tfrac{1}{2}\ddot{x}_k t_{inb}^2, \\ \dot{x} &= \dot{x}_{jk} + \ddot{x}_k t_{inb}, \\ \ddot{x} &= \ddot{x}_k, \end{aligned} \tag{7.48}$$

en donde las cantidades $\dot{x}_{jk}$, $\ddot{x}_k$, t_j y t_{jk} se calcularon en el tiempo de planeación, de la misma forma que en el caso del espacio de articulación.

Por último, esta trayectoria cartesiana (χ, $\dot{\chi}$ y $\ddot{\chi}$) debe convertirse en cantidades equivalentes en espacio de articulación. Una solución analítica completa para este problema utilizaría la cinemática inversa para calcular las posiciones de las articulaciones, el jacobiano inverso para las velocidades y el jacobiano inverso más su derivada para las aceleraciones [5]. Una forma más sencilla que se utiliza mucho en la práctica es la siguiente: en la proporción de actualización de ruta convertimos χ en su representación de trama equivalente, ${}^{S}_{G}T$. Después utilizamos la rutina RESOLVER (vea la sección 4.8)

para calcular el vector requerido de los ángulos de articulación Θ. Luego utilizamos diferenciación numérica para calcular $\dot{\Theta}$ y $\ddot{\Theta}$.[3] Por consecuencia, el algoritmo es

$$\begin{aligned} \chi &\rightarrow {}^{S}_{G}T, \\ \Theta(t) &= RESOLVER({}^{S}_{G}T), \\ \dot{\Theta}(t) &= \frac{\Theta(t) - \Theta(t - \delta t)}{\delta t}, \\ \ddot{\Theta}(t) &= \frac{\dot{\Theta}(t) - \dot{\Theta}(t - \delta t)}{\delta t}. \end{aligned} \tag{7.49}$$

Entonces se proporcionan Θ, $\dot{\Theta}$ y $\ddot{\Theta}$ al sistema de control del manipulador.

7.7 DESCRIPCIÓN DE RUTAS CON UN LENGUAJE DE PROGRAMACIÓN DE ROBOTS

En el capítulo 12 hablaremos más sobre los **lenguajes de programación de robots**. Aquí ilustraremos cómo podrían especificarse en un lenguaje de robot varios tipos de rutas que hemos visto en este capítulo. En estos ejemplo utilizaremos la sintaxis de **AL**, un lenguaje de programación de robots desarrollado en la Universidad de Stanford [6].

Los símbolos A, B, C y D representan variables de tipo "trama" en los ejemplos de lenguaje AL que se muestran a continuación. Estas tramas especifican puntos de ruta que supondremos se han enseñado o se han descrito textualmente para el sistema. Suponga que el manipulador empieza en la posición A. Para mover el manipulador (ARM) en modo de espacio de articulación a lo largo de rutas de mezcla parabólica-lineal, podríamos decir

```
move ARM to C with duration = 3*seconds;
```

Para movernos a la misma posición y orientación en una línea recta podríamos decir

```
move ARM to C linearly with duration = 3*seconds;
```

en donde la palabra clave "linearly" denota que se va a usar el movimiento cartesiano en línea recta. Si la duración (duration) no es importante, se puede omitir esta especificación y el sistema utilizará una velocidad predeterminada; esto es,

```
move ARM to C;
```

Si queremos agregar un punto vía, escribimos

```
move ARM to C via B;
```

o podría especificarse todo un conjunto de puntos vía mediante

```
move ARM to C via B,A,D;
```

Observe que en

```
move ARM to C via B with duration = 6*seconds;
```

se proporciona la duración para todo el movimiento. El sistema decide cómo dividir esta duración entre los dos segmentos. En AL es posible especificar la duración de un solo segmento; por ejemplo, mediante

```
move ARM to C via B where duration = 3*seconds;
```

El primer segmento que conduzca al punto B tendrá una duración de 3 segundos.

7.8 PLANEACIÓN DE RUTAS CUANDO SE USA EL MODELO DINÁMICO

Generalmente, cuando se planean rutas utilizamos una aceleración predeterminada o máxima en cada punto de mezcla. En realidad, la cantidad de aceleración que el manipulador

[3] Esta diferenciación puede realizarse de manera no causal para las rutas previamente planeadas, produciendo $\dot{\Theta}$ y $\ddot{\Theta}$ de mejor calidad. Además, muchos sistemas de control no requieren una entrada $\ddot{\Theta}$, por lo que no se calcularía

es capaz de producir en cualquier instante es una función de la dinámica del brazo y de los límites del actuador. La mayoría de los actuadores no se caracterizan por un momento de torsión o una aceleración máxima fija, sino por una curva momento de torsión-velocidad.

Cuando planeamos una ruta suponiendo que hay una aceleración máxima en cada articulación o a lo largo de cada grado de libertad, estamos haciendo una gran simplificación. Para tener cuidado de no exceder las capacidades reales del dispositivo, esta aceleración máxima debe elegirse de manera conservadora. Por lo tanto, no estamos utilizando por completo las capacidades de velocidad del manipulador en rutas planeadas por los métodos que presentamos en este capítulo.

Podríamos hacer la siguiente pregunta: dada una ruta espacial deseada del efector final, encuentre la información de temporización (que convierte la descripción de una ruta espacial en una trayectoria) de tal forma que el manipulador llegue al punto de destino en un tiempo mínimo. Dichos problemas se han resuelto mediante métodos numéricos [7, 8]. La solución toma en cuenta la dinámica del cuerpo rígido y las curvas de restricción de velocidad-momento de torsión del actuador.

7.9 PLANEACIÓN DE RUTAS SIN COLISIONES

Sería muy conveniente simplemente decir al sistema de robot cuál es el punto de destino deseado del movimiento del manipulador y dejar que el sistema determine en dónde y cuántos puntos vía se requieren para llegar al destino sin que el manipulador choque contra algún obstáculo. Para ello, el sistema debe tener modelos del manipulador, del área de trabajo y de todos los posibles obstáculos en el área. Incluso podría haber otro manipulador trabajando en la misma área, en cuyo caso cada brazo tendría que considerarse como un obstáculo móvil para el otro.

Los sistemas que planean rutas sin colisiones no están disponibles comercialmente. La investigación en esta área ha permitido descubrir dos técnicas rivales principales, junto con diversas variaciones y combinaciones de las mismas. Uno de los métodos resuelve el problema formando una representación en gráficos conectados del espacio libre y después buscando en el gráfico una ruta sin colisiones [9-11, 17, 18]. Desafortunadamente estas técnicas tienen una complejidad exponencial en el número de articulaciones en el dispositivo. El segundo método se basa en la creación de campos artificales de potencial alrededor de los obstáculos que hacen que el o los manipuladores eviten los obstáculos mientras son atraídos hacia un polo de atracción artificial en el punto de destino [12]. Por desgracia estos métodos casi siempre tienen una vista local del entorno y pueden quedar "atascados" en los mínimos locales del campo artificial.

BIBLIOGRAFÍA

[1] R. P. Paul y H. Zong, "Robot Motion Trajectory Specification and Generation", 2o. Simposio Internacional en Investigación Robótica, Kyoto, Japón, agosto de 1984.

[2] R. Taylor, "Planning and Execution of Straight Line Manipulator Trajectories", en *Robot Motion*, Brady y cols., Editores, MIT Press, Cambridge, MA, 1983.

[3] C. DeBoor, *A Practical Guide to Splines*, Springer-Verlag, Nueva York, 1978

[4] D. Rogers y J. A. Adams, *Mathematical Elements for Computer Graphics*, McGraw-Hill, Nueva York, 1976.

[5] B. Gorla y M. Renaud, *Robots Manipulateurs*. Cepadues-Editions, Toulouse, 1984.

[6] R. Goldman, *Design of an Interactive Manipulator Programming Environment*, UMI Research Press, Ann Arbor, MI, 1985.

[7] J. Bobrow, S. Dubowsky y J. Gibson, "On the Optimal Control of Robotic Manipulators with Actuator Constraints", *Proceedings of the American Control Conference*, junio de 1983.

[8] K. Shin y N. McKay, "Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints", *IEEE Transactions on Automatic Control*, junio de 1985.

[9] T. Lozano-Perez, "Spatial Planning: A Configuration Space Approach", AI Memo 605, Laboratorio de Inteligencia Artificial del MIT, Cambridge, MA, 1980.

[10] T. Lozano-Perez, "A Simple Motion Planning Algorithm for General Robot Manipulators", *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 3, junio de 1987.

[11] R. Brooks, "Solving the Find-Path Problem by Good Representation of Free Space", *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13:190-197, 1983.

[12] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", *The International Journal of Robotics Research*, Vol. 5, No. 1, primavera de 1986.

[13] R. P. Paul, "Robot Manipulators: Mathematics, Programming and Control", MIT Press, Cambridge, MA, 1981.

[14] R. Castain y R. P. Paul, "An Online Dynamic Trajectory Generator", *The International Journal of Robotics Research*, Vol. 3, 1984.

[15] C. S. Lin y P. R. Chang, "Joint Trajectory of Mechanical Manipulators for Cartesian Path Approximation", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-13, 1983.

[16] C. S. Lin, P.R. Chang y J. Y .S. Luh, "Formulation and Optimization of Cubic Polinomial Joint Trajectories for Industrial Robots", *IEEE Transactions on Automatic Control*, Vol. AC-28, 1983.

[17] L. Kavraki, P. Svestka, J. C. Latombe y M. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces", *IEEE Transactions on Robotics and Automation*, 12(4): 566-580, 1996.

[18] J. Barraquand, L. Kavraki, J. C. Latombe, T. Y. Li, R. Motwani y P. Raghavan, "A Random Sampling Scheme for Path Planning", *International Journal of Robotics Research*, 16(6): 759-774, 1977.

EJERCICIOS

7.1 [8] ¿Cuántos polinomios cúbicos individuales se calculan cuando un robot de seis articulaciones se mueve a lo largo de una ruta de trazador cúbico a través de dos puntos vía y se detiene en un punto de destino? ¿Cuántos coeficientes se almacenan para describir estos polinomios cúbicos?

7.2 [13] Un robot de un solo vínculo con una articulación giratoria está sin movimiento, con $\theta = -5°$. Se desea mover la articulación de una manera uniforme hasta $\theta = 80°$ en 4 segundos y que se detenga uniformemente. Encuentre los coeficientes de un polinomio cúbico que logre este movimiento y lleve el brazo al punto de destino, en donde quedará inmóvil. Trace la posición, la velocidad y la aceleración de la articulación como una función del tiempo.

7.3 [14] Un robot de un solo vínculo con una articulación giratoria está sin movimiento, con $\theta = -5°$. Se desea mover la articulación de una manera uniforme hasta $\theta = 80°$ en 4 segundos y que se detenga uniformemente. Calcule los parámetros correspondientes de una trayectoria lineal con mezclas parabólicas. Trace la posición, la velocidad y la aceleración de la articulación como una función del tiempo.

7.4 [30] Escriba una rutina de software para una planeación de ruta que implemente las ecuaciones (7.24) a (7.28) de una forma general para las rutas descritas por un número arbitrario de puntos de ruta. Esta rutina podría utilizarse para resolver el ejemplo 7.4.

7.5 [18] Bosqueje gráficos de posición, velocidad y aceleración para el trazador de aceleración continua de dos segmentos dado en el ejemplo 7.2. Haga los bosquejos para una articulación en la que $\theta_0 = 5.0°$, $\theta_v = 15.0°$, $\theta_g = 40.0°$ y que cada segmento dure 1.0 segundos.

7.6 [18] Bosqueje gráficos de posición, velocidad y aceleración para un trazador de dos segmentos, en donde cada segmento sea un polinomio cúbico, utilizando los coeficientes que se proporcionan en la ecuación (7.11). Haga los bosquejos para una articulación en la que $\theta_0 = 5.0°$ para el punto inicial, $\theta_v = 15.0°$ sea un punto vía y $\theta_g = 40.0°$ sea el punto de destino. Suponga que cada segmento tiene una duración de 1.0 segundo y que la velocidad en el punto vía es de 17.5 grados/segundo.

7.7 [20] Calcule $\dot{\theta}_{12}$, $\dot{\theta}_{23}$, t_1, t_2 y t_3 para una trazador lineal de dos segmentos con mezclas parabólicas. [Use las ecuaciones (7.24) a (7.28)]. Para esta articulación, $\theta_1 = 5.0°$, $\theta_2 = 15.0°$, $\theta_3 = 40.0°$. Suponga que $t_{d12} = t_{d23} = 1.0$ segundos y que la aceleración predeterminada a utilizar durante las mezclas es de 80 grados/seg^2. Bosqueje trazos de posición, velocidad y aceleración en θ.

7.8 [18] Bosqueje gráficos de posición, velocidad y aceleración para el trazador de dos segmentos y aceleración continua que se proporciona en el ejemplo 7.2. Haga los bosquejos para una articulación en la que $\theta_0 = 5.0°$, $\theta_v = 15.0°$, $\theta_g = -10.0°$ y que cada segmento dure 2.0 segundos.

7.9 [18] Bosqueje gráficos de posición, velocidad y aceleración para un trazador de dos segmentos en el que cada segmento sea un polinomio cúbico, utilizando los coeficientes que se proporcionan en la ecuación (7.11). Haga los bosquejos para una articulación en la que $\theta_0 = 5.0°$ para el punto inicial, $\theta_v = 15.0°$ sea un punto vía y $\theta_g = -10.0°$ sea el punto de destino. Suponga que cada segmento tiene una duración de 2.0 segundos y que la velocidad en el punto vía es de 0.0 grados/segundo.

7.10 [20] Calcule $\dot{\theta}_{12}$, $\dot{\theta}_{23}$, t_1, t_2 y t_3 para un trazador lineal de dos segmentos con mezclas parabólicas. [Use las ecuaciones (7.24) a (7.28)]. Para esta articulación, $\theta_1 = 5.0°$, $\theta_2 = 15.0°$, $\theta_3 = -10.0°$. Suponga que $t_{d12} = t_{d23} = 2.0$ segundos y que la aceleración predeterminada a utilizar durante las mezclas es de 60 grados/seg^2. Bosqueje trazos de posición, velocidad y aceleración en θ.

7.11 [6] Proporcione la representación ${}^S\chi_G$ de la posición y la orientación cartesianas de 6×1 que sea equivalente a ${}^S_G T$, en donde ${}^S_G R = ROT(\hat{Z}, 30°)$ y ${}^S P_{GORG} = [10.0\ \ 20.0\ \ 30.0]^T$.

7.12 [6] Proporcione la ${}^S_G T$ que sea equivalente a la representación de la posición y la orientación cartesianas de 6×1, representada por ${}^S\chi_G = [5.0\ \ -20.0\ \ 10.0\ \ 45.0\ \ 0.0\ \ 0.0]^T$.

7.13 [30] Escriba un programa que utilice las ecuaciones dinámicas de la sección 6.7 (el manipulador planar de dos vínculos) para calcular el historial en tiempo de los momentos de torsión necesarios para mover el brazo a lo largo de la trayectoria del ejercicio 7.8. ¿Cuáles son los máximos momentos de torsión requeridos y en dónde ocurren a lo largo de la trayectoria?

7.14 [32] Escriba un programa que utilice las ecuaciones dinámicas de la sección 6.7 (el manipulador planar de dos vínculos) para calcular el historial en tiempo de los momentos de torsión necesarios para mover el brazo a lo largo de la trayectoria del ejercicio 7.8. Realice trazos separados de los momentos de torsión requeridos debido a la inercia, los términos de la velocidad y la gravedad.

7.15 [22] Haga el ejemplo 7.2 cuando $t_{f1} \neq t_{f2}$.

7.16 [25] Deseamos mover una sola articulación de θ_0 a θ_f que empiece desde una posición inmóvil y termine en una posición inmóvil, en el tiempo t_f. Se proporcionan los valores de θ_0 y θ_f, pero deseamos calcular t_f de tal forma que $||\dot{\theta}(t)|| < \dot{\theta}_{máx}$ y $||\ddot{\theta}(t)|| < \ddot{\theta}_{máx}$ para todos los valores de t en donde a $\dot{\theta}_{máx}$ y $\ddot{\theta}_{máx}$ se les asignen constantes positivas. Utilice un solo segmento cúbico y proporcione una expresión para t_f y para los coeficientes del polinomio cúbico.

7.17 [10] Una trayectoria cúbica individual se da mediante

$$\theta(t) = 10 + 90t^2 - 60t^3$$

y se utiliza sobre el intervalo de tiempo desde $t = 0$ hasta $t = 1$. ¿Cuáles son las posiciones, las velocidades y las aceleraciones iniciales y finales?

7.18 [12] Una trayectoria cúbica individual se da mediante

$$\theta(t) = 10 + 90t^2 - 60t^3$$

y se utiliza sobre el intervalo de $t = 0$ hasta $t = 2$. ¿Cuáles son las posiciones, las velocidades y las aceleraciones iniciales y finales?

7.19 [13] Una trayectoria cúbica individual se da mediante

$$\theta(t) = 10 + 5t + 70t^2 - 45t^3$$

y se utiliza sobre el intervalo de $t = 0$ hasta $t = 1$. ¿Cuáles son las posiciones, las velocidades y las aceleraciones iniciales y finales?

7.20 [12] Una trayectoria cúbica individual se da mediante

$$\theta(t) = 10 + 5t + 70t^2 - 45t^3$$

y se utiliza sobre el intervalo de $t = 0$ hasta $t = 2$. ¿Cuáles son las posiciones, las velocidades y las aceleraciones iniciales y finales?

EJERCICIO DE PROGRAMACIÓN (PARTE 7)

1. Escriba un sistema de planeación de rutas tipo trazador cúbico en espacio de articulación. Una rutina que su sistema deberá incluir es

```
Procedure COEFCUB (VAR th0, thf, thpunto0, thpuntof: real;
VAR cc: vec4);
```

en donde

$$\begin{aligned} \text{th0} &= \text{posición inicial de } \theta \text{ en la parte inicial del segmento,} \\ \text{thf} &= \text{posición final de } \theta \text{ en la parte final del segmento,} \\ \text{thpunto0} &= \text{velocidad inicial del segmento,} \\ \text{thpuntof} &= \text{velocidad final del segmento.} \end{aligned}$$

Estas cuatro cantidades son entradas y “cc”, es un arreglo de los cuatro coeficientes cúbicos, es la salida.

Su programa deberá aceptar (cuando menos) cinco especificaciones de puntos vía, en la forma de trama herramienta $\{T\}$ respecto a la trama estación $\{S\}$, en el formato común de usuario: (x, y, ϕ). Para facilitarnos la vida, todos los segmentos tendrán la misma duración. Su sistema deberá despejar los coeficientes de los polinomios cúbicos usando cierta heurística razonable para asignar velocidades de articulación en los puntos vía. *Sugerencia*: vea la opción 2 en la sección 7.3.

2. Escriba un sistema generador de rutas que calcule una trayectoria en el espacio de articulación basada en conjuntos de coeficientes cúbicos para cada segmento. Debe ser capaz de generar la ruta de varios segmentos que usted planeó en el problema 1. El usuario deberá especificar una duración para los segmentos. Deberá producir información sobre la posición, la velocidad y la aceleración a la frecuencia de actualización de ruta, que también deberá especificar el usuario.

3. El manipulador es el mismo de tres vínculos que se usó anteriormente. Las definiciones de las tramas $\{T\}$ y $\{S\}$ son las mismas que antes:

$$^{W}_{T}T = [x\ y\ \theta] = [0.1\ \ 0.2\ \ 30.0],$$

$$^{B}_{S}T = [x\ y\ \theta] = [0.0\ \ 0.0\ \ 0.0].$$

Utilizando una duración de 3.0 segundos por segmento, planee y ejecute la ruta que empieza con el manipulador en la posición

$$[x_1\ y_1\ \phi_1] = [0.758\ \ 0.173\ \ 0.0],$$

que se desplaza a través de los puntos vía

$$[x_2\ y_2\ \phi_2] = [0.6\ \ -0.3\ \ 45.0]$$

y

$$[x_3\ y_3\ \phi_3] = [-0.4\ \ 0.3\ \ 120.0],$$

y que termina en el punto de destino (en este caso, el mismo que el punto inicial)

$$[x_4\ y_4\ \phi_4] = [0.758\ \ 0.173\ \ 0.0].$$

Use una frecuencia de actualización de ruta de 40 Hz, pero imprima la posición sólo cada 0.2 segundos. Imprima las posiciones en términos cartesianos para usuario. No tiene que imprimir velocidades o aceleraciones, aunque sería conveniente hacerlo.

EJERCICIO MATLAB 7

El objetivo de este ejercicio es implementar ecuaciones polinomiales de generación de trayectorias en espacio de articulación para una sola articulación. (Para varias articulaciones se requerirían *n* aplicaciones del resultado). Escriba un programa en MATLAB para implementar la generación de trayectorias en espacio de articulación para los tres casos siguientes. Reporte sus resultados para las asignaciones específicas dadas; en cada caso, proporcione las funciones polinomiales para el ángulo de articulación, la velocidad angular, la aceleración angular y la derivada en el tiempo de la aceleración angular. Para cada caso trace los resultados. (Ordene los trazos verticalmente con ángulo, velocidad, aceleración y después la derivada de la aceleración, todo con la misma escala de tiempo; compruebe la función *subplot* de MATLAB para lograr esto.) No sólo trace los resultados, comente algo; ¿sus resultados tienen sentido? He aquí los tres casos:

a) *Polinomio de tercer orden.* Haga que la velocidad angular sea cero al principio y al final. Se dan $\theta_s = 120°$ (inicio), $\theta_f = 60°$ (final) y $t_f = 1$ segundo.
b) *Polinomio de quinto orden.* Haga que la velocidad angular y la aceleración sean cero al principio y al final. Se dan $\theta_s = 120°$, $\theta_f = 60°$ y $t_f = 1$ segundo. Compare sus resultados (funciones y trazos) con este mismo ejemplo, pero usando un polinomio de tercer orden, como en el problema (a).
c) *Dos polinomios de tercer orden con punto vía*. Haga que la velocidad angular sea cero al principio y al final. No haga que la velocidad angular sea cero en el punto vía; se deben igualar la velocidad y la aceleración en este punto para que los dos polinomios se encuentren en ese punto en el tiempo. Demuestre que se satisface esta condición. Se dan $\theta_s = 60°$ (inicio), $\theta_v = 120°$ (vía), $\theta_f = 30°$ (final) y $t_1 = t_2 = 1$ segundo (intervalos relativos; por ejemplo, $t_f = 2$ segundos).
d) Compruebe los resultados de (a) y (b) mediante la Caja de Herramientas de Robótica MATLAB de Corke. Pruebe la función *jtraj()*.

CAPÍTULO 8

Diseño del mecanismo del manipulador

8.1 INTRODUCCIÓN

En capítulos anteriores hemos visto que la estructura específica de un manipulador ejerce influencia sobre el análisis cinemático y dinámico. Por ejemplo, ciertas configuraciones cinemáticas serán fáciles de resolver; otras no tendrán solución cinemática de forma cerrada. De igual forma, la complejidad de las ecuaciones dinámicas puede variar considerablemente con la configuración cinemática y la distribución de la masa de los vínculos. En los siguientes capítulos veremos que el control del manipulador no sólo depende de la dinámica de cuerpo rígido, sino también de la fricción y la flexibilidad de los sistemas controladores.

Las tareas que pueda realizar un manipulador también variarán considerablemente con el diseño específico. Aunque generalmente hemos tratado con el manipulador del robot como una entidad abstracta, su rendimiento está limitado en última instancia por factores pragmáticos tales como la capacidad de carga, la velocidad, el tamaño del espacio de trabajo y la repetitividad. Para ciertas aplicaciones, el tamaño total del manipulador, su peso, su consumo de energía y su costo serán factores de consideración.

En este capítulo hablaremos de algunas de las cuestiones involucradas en el diseño del manipulador. En general, los métodos de diseño e incluso la evaluación de un diseño terminado son temas parcialmente subjetivos. Es difícil limitar el espectro de opciones de diseño con muchas reglas inflexibles.

Los elementos de un sistema de robot pueden dividirse en cuatro categorías:

1. El manipulador, incluyendo sus sensores internos o **propioceptivos**.
2. El efector final, o **herramientas en el extremo del brazo**.
3. Sensores y efectores externos, como sistemas de visión y alimentadores de piezas.
4. El controlador.

La amplitud de disciplinas de ingeniería que se abarcan nos obliga a restringir nuestra atención únicamente al diseño del manipulador en sí.

Para desarrollar el diseño de un manipulador, empezamos examinando los factores que probablemente tengan el mayor efecto general en el diseño y después consideramos las preguntas más detalladas. No obstante, en el fondo el diseño de un manipulador es un proceso iterativo. Es muy probable que los problemas que surgen al resolver un detalle de diseño nos obliguen a reformular las decisiones anteriores de diseño de un mayor nivel.

8.2 DISEÑO BASADO EN LOS REQUERIMIENTOS DE LA TAREA

Aunque los robots son nominalmente máquinas "universalmente programables", capaces de realizar una amplia variedad de tareas, factores prácticos y económicos dictan que se deben diseñar distintos manipuladores para tipos específicos de tareas. Por ejemplo, los grandes robots capaces de manejar cargas útiles de cientos de libras no tienen generalmente la capacidad de insertar componentes electrónicos en tableros de circuitos. Como veremos, no sólo el tamaño, sino también el número de articulaciones, el orden de las articulaciones y los tipos de accionamiento, detección y control variarán enormemente con el tipo de tarea a realizar.

Número de grados de libertad

El número de grados de libertad en un manipulador debe concordar con el número requerido por la tarea. No todas las tareas requieren de seis grados completos de libertad.

La circunstancia más común de este tipo ocurre cuando el efector final tiene un eje de simetría. La figura 8.1 muestra a un manipulador colocando una herramienta de esmerilado en dos formas distintas. En este caso, la orientación de la herramienta respecto al eje de la misma, $\hat{Z}_T$, es irrelevante, ya que la muela está girando a varios cientos de RPM. Al decir que podemos posicionar este robot 6-DOF (seis grados de libertad) en una infinidad de formas para esta tarea (la rotación sobre $\hat{Z}_T$ es una variable libre), estamos diciendo que el robot es **redundante** para esta tarea. La soldadura con arco, la soldadura por puntos, el desbarbado, el encolado y el pulido son otros ejemplos de tareas que a menudo emplean efectores finales con cuando menos un eje de simetría.

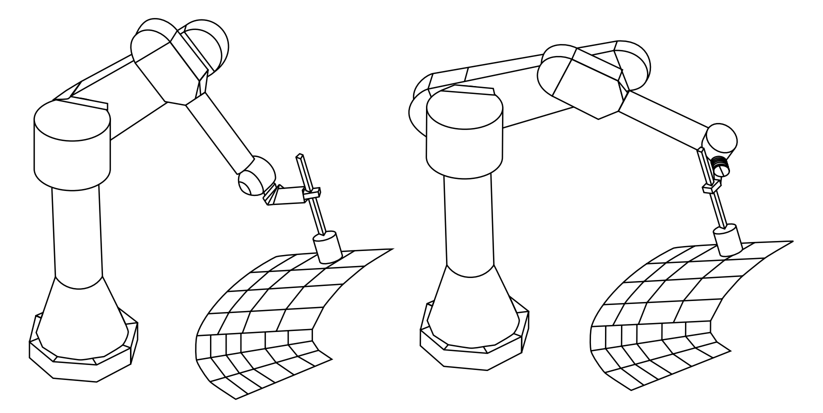


FIGURA 8.1: Un manipulador 6-DOF con una herramienta simétrica contiene un grado redundante de libertad.

Al analizar la situación de la herramienta simétrica, algunas veces es útil imaginar una *articulación ficticia* cuyo eje esté sobre el eje de simetría. Al colocar un efector final en una ubicación específica, necesitamos un total de seis grados de libertad. Como una de estas seis es nuestra herramienta ficticia, el manipulador no necesita tener más de cinco grados de libertad. Si se utilizara un robot 5-DOF en la aplicación de la figura 8.1, entonces regresaríamos al caso común en el que sólo hay un número finito de soluciones distintas disponibles para posicionar la herramienta. Un gran porcentaje de los robots industriales existentes son 5-DOF, reconociendo el relativo predominio de las aplicaciones con herramientas simétricas.

Algunas tareas se llevan a cabo en dominios que, fundamentalmente, tienen menos de seis grados de libertad. La colocación de componentes en tableros de circuitos es un ejemplo común de esto. Por lo general, los tableros de circuitos son planares y contienen piezas de varias alturas. Para posicionar piezas en una superficie planar se requieren tres grados de libertad (x, y y θ); para poder levantar e insertar las piezas se agrega un cuarto movimiento normal al plano (z).

Los robots con menos de seis grados de libertad pueden también realizar tareas en las que cierto tipo de dispositivo de colocamiento activo presenta las piezas. Por ejemplo, en la soldadura de tuberías, una plataforma de inclinación/giro, como la mostrada en la figura 8.2, es la que comúnmente presenta las piezas a soldar. Al contar el número de grados de libertad entre los tubos y el efector final, la plataforma de inclinación/giro representa dos. Esto, junto con el hecho de que la soldadura de arco es una tarea de herramienta simétrica, significa que en teoría podría usarse un manipulador 3-DOF. En la práctica, la necesidad de evitar colisiones con la pieza de trabajo es lo que al final hace que se utilice un robot con más grados de libertad.

Las piezas con un eje de simetría también reducen los grados requeridos de libertad para el manipulador. Por ejemplo, las piezas cilíndricas pueden en muchos casos recogerse e insertarse independientemente de la orientación del sujetador, respecto al eje del cilindro. No obstante, hay que tener en cuenta que, una vez sujeta la pieza, la orientación de ésta sobre su eje simétrico no debe importar en *todas* las operaciones subsiguientes, ya que no está garantizada su orientación.

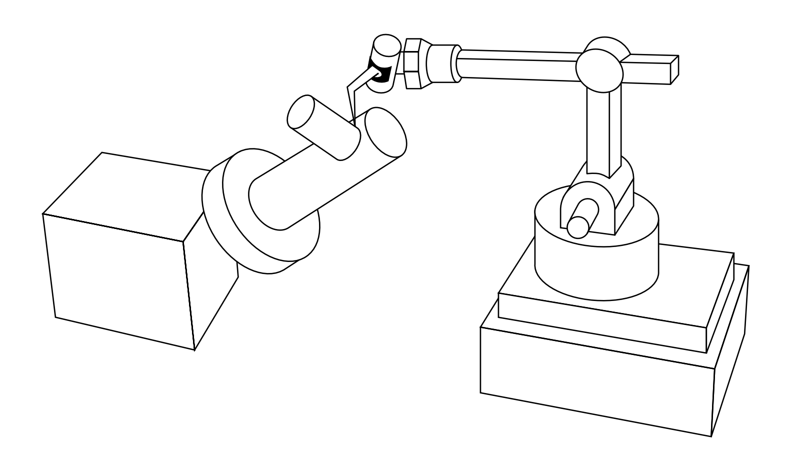


FIGURA 8.2: Una plataforma de inclinación/giro proporciona dos grados de libertad al sistema manipulador general.

Espacio de trabajo

Para realizar tareas, un manipulador tiene que alcanzar determinado número de piezas de trabajo o dispositivos fijos. En algunos casos estos objetos pueden colocarse, según sea necesario, para adaptarse al espacio de trabajo del manipulador. En otros casos puede instalarse un robot en un entorno fijo con requerimientos de un espacio de trabajo rígido. Al **espacio de trabajo** también se le llama algunas veces **volumen de trabajo** o **envoltura de trabajo**.

La escala total de la tarea establece el espacio de trabajo requerido del manipulador. En algunos casos, los detalles de la forma del espacio de trabajo y la ubicación de las singularidades del mismo serán consideraciones importantes.

La intrusión del manipulador en el espacio de trabajo puede algunas veces ser un factor. Dependiendo del diseño cinemático, la operación de un manipulador en una aplicación dada podría requerir más o menos espacio alrededor de los dispositivos fijos para poder evitar colisiones. Los entornos restringidos pueden afectar la elección de la configuración cinemática.

Capacidad de carga

La **capacidad de carga** de un manipulador depende del tamaño de sus miembros estructurales, del sistema de transmisión de energía y de los actuadores. La carga colocada en los actuadores y en el sistema de control es una función de la configuración del robot, del porcentaje de tiempo que se soporta una carga y de la carga dinámica debida a las fuerzas relacionadas con la inercia y la velocidad.

Velocidad

Un objetivo obvio en el diseño ha sido tratar de crear manipuladores cada vez más rápidos. La alta velocidad ofrece ventajas en muchas aplicaciones en las que una solución robótica propuesta debe competir en términos económicos con trabajadores humanos o sistemas automáticos. No obstante, para algunas aplicaciones el proceso en sí es el que limita la velocidad, y no el manipulador. Éste es el caso con muchas aplicaciones de soldadura y pintura con pistola de aire.

Hay una distinción importante entre la máxima velocidad del efector final y el **tiempo de ciclo** total para una tarea específica. Para las aplicaciones que involucran sujetar y colocar, el manipulador debe acelerar y desacelerar hacia y desde las ubicaciones de *sujetar* y *colocar* dentro de ciertos rangos de precisión en la posición. A menudo, las fases de aceleración y desaceleración ocupan la mayor parte del tiempo de ciclo. Por consecuencia, la capacidad de aceleración es muy importante, no sólo la velocidad pico.

Repetitividad y precisión

Es muy costoso lograr una alta repetitividad y precisión, aunque es algo muy conveniente en el diseño de cualquier manipulador. Por ejemplo, sería absurdo diseñar un robot de pintura con pistola de aire para ser preciso dentro de un margen de 0.001 pulgadas cuando el diámetro del punto que se va a pintar es de 8 ± 2 pulgadas. En buena medida, la precisión de un modelo específico de robot industrial depende de los detalles de su fabricación, y no de su diseño. La alta precisión se logra mediante un buen conocimiento de los parámetros del vínculo (y otros más). Para hacer esto posible se necesitan mediciones precisas después de la fabricación, o una cuidadosa atención a las tolerancias durante la fabricación.

8.3 CONFIGURACIÓN CINEMÁTICA

Una vez que se ha decidido el número requerido de grados de libertad, debe seleccionarse una configuración específica de articulaciones para realizar esos grados de libertad. Para los vínculos cinemáticos seriales, el número de articulaciones es igual al número requerido de grados de libertad. La mayoría de los manipuladores están diseñados de manera que las últimas $n - 3$ articulaciones orienten al efector final y tengan ejes que se intersequen en el **punto de la muñeca**, y las primeras tres articulaciones colocan este punto de la muñeca. Se puede decir que los manipuladores con este diseño están compuestos de una **estructura de posicionamiento**, seguida de una **estructura de orientación** o **muñeca**. Como vimos en el capítulo 4, estos manipuladores siempre tienen soluciones cinemáticas de forma cerrada. Aunque existen otras configuraciones que poseen soluciones cinemáticas de forma cerrada, casi todos los manipuladores industriales pertenecen a esta clase de mecanismos con **partición de muñeca**. Además, la estructura de posicionamiento está, casi sin excepciones, diseñada para ser cinemáticamente simple, con torsiones de vínculos iguales a 0 o ±90 grados, y con muchas de las longitudes de vínculos y desplazamientos iguales a cero.

Se ha convertido en una costumbre clasificar los manipuladores de la clase simple cinemática, con partición de muñeca, de acuerdo con el diseño de sus primeras tres articulaciones (la estructura de posicionamiento). En los siguientes párrafos describiremos brevemente la más común de estas clasificaciones.

Cartesiano

Un **manipulador cartesiano** es tal vez la configuración más simple. Como se muestra en la figura 8.3, las articulaciones, de la 1 a la 3, son prismáticas, mutuamente ortogonales y corresponden a las direcciones cartesianas $\hat{X}$, $\hat{Y}$ y $\hat{Z}$. La solución cinemática inversa para esta configuración es trivial.

Esta configuración produce robots con estructuras muy rígidas. Como consecuencia pueden construirse robots bastante grandes. Estos grandes robots, conocidos comúnmente como **robots de puente transversal**, se asemejan a las grúas corredizas aéreas. Los robots de puente transversal algunas veces manipulan automóviles completos o inspeccionan aeronaves completas.

Las otras ventajas de los manipuladores cartesianos se deben al hecho de que las primeras tres articulaciones están *desacopladas*. Esto facilita su diseño y evita las singularidades cinemáticas debidas a las primeras tres articulaciones.

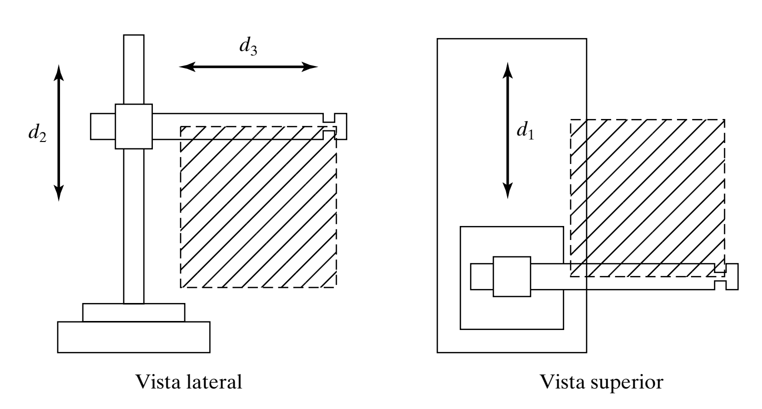


FIGURA 8.3: Un manipulador articulado.

Su principal desventaja es que todos los alimentadores y dispositivos fijos asociados con una aplicación deben encontrarse "dentro" del robot. En consecuencia, las celdas de trabajo de las aplicaciones para los robots cartesianos se vuelven muy dependientes de la máquina. El tamaño de la estructura de soporte del robot limita el tamaño y la colocación de los dispositivos fijos y los sensores. Estas limitaciones hacen extremadamente difícil el proceso de modernizar los robots cartesianos para adaptarlos a las celdas de trabajo existentes.

Articulado

La figura 8.4 muestra un **manipulador articulado**, algunas veces llamado también manipulador de **codo** o **antropomórfico**. Un manipulador de este tipo generalmente consiste en dos articulaciones tipo "hombro" (una para girar sobre un eje vertical y otra para elevarse en el plano horizontal), una articulación tipo "codo" (cuyo eje es normalmente paralelo a la articulación de elevación del hombro) y dos o tres articulaciones de muñeca en el extremo del manipulador. Tanto el PUMA 560 como el Motoman L-3, que estudiamos en capítulos anteriores, se encuentran dentro de esta categoría.

Los robots articulados minimizan la intrusión de la estructura del manipulador en el espacio de trabajo, haciéndolos capaces de llegar a espacios confinados. Requieren de una estructura total mucho menor que los robots cartesianos, lo que los hace menos costosos para aplicaciones que requieren de espacios de trabajo más pequeños.

SCARA

La configuración **SCARA**,[1] como la que se muestra en la figura 8.5, tiene tres articulaciones angulares paralelas (que le permiten moverse y orientarse en un plano), con una cuarta articulación prismática para mover el efector final de forma normal al plano. La ventaja principal es que las primeras tres articulaciones no tienen que soportar el peso del manipulador o de la carga. Además, el vínculo 0 puede alojar fácilmente los actuadores para las primeras dos articulaciones. Los actuadores pueden hacerse bastante grandes, por lo que el robot puede moverse con gran rapidez. Por ejemplo, un manipulador SCARA

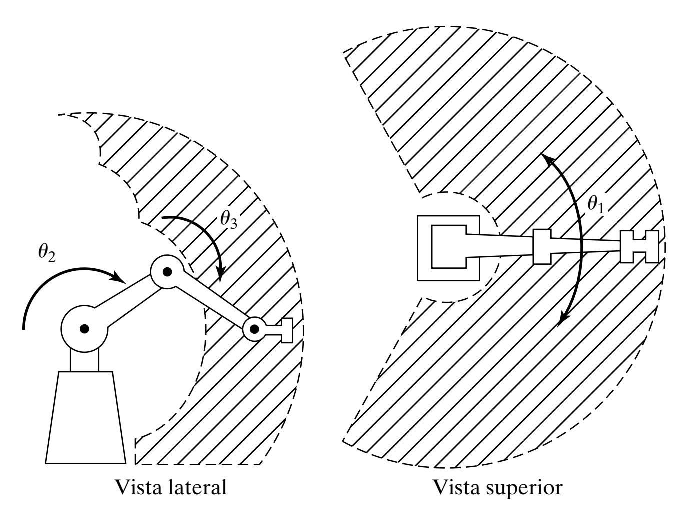


FIGURA 8.4: Un manipulador articulado.

[1] SCARA significa "brazo de robot con montaje selectivamente obediente", por sus siglas en inglés.

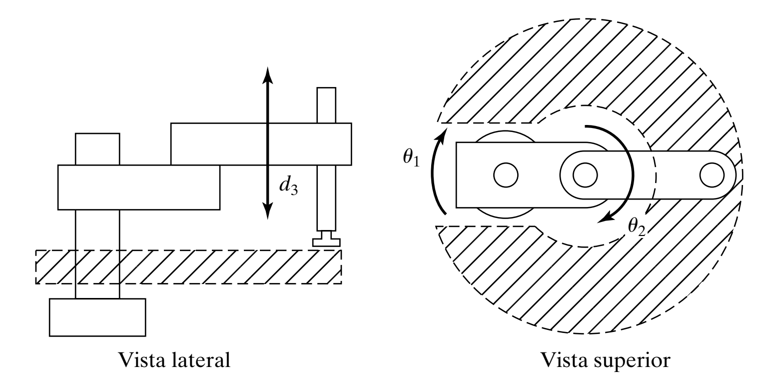


FIGURA 8.5: Un manipulador SCARA.

puede moverse a una velocidad de hasta 30 pies por segundo, casi 10 veces más rápido que la mayoría de los robots industriales articulados [1]. Esta configuración se adapta mejor a las tareas en entornos planos.

Esférico

La configuración esférica de la figuras 8.6 tiene muchas similitudes con el manipulador articulado, sólo que la articulación tipo codo se remplaza por una articulación prismática. Este diseño se adapta mejor que el diseño tipo codo en algunas aplicaciones. El vínculo que se mueve prismáticamente podría plegarse como telescopio (o incluso "salirse por detrás") al retractarse.

Cilíndrico

Los manipuladores **cilíndricos** (figura 8.7) consisten de una articulación prismática para trasladar el brazo en forma vertical, una articulación angular con un eje vertical, otra articulación prismática ortogonal al eje de la articulación angular y, finalmente, una muñeca de cierto tipo.

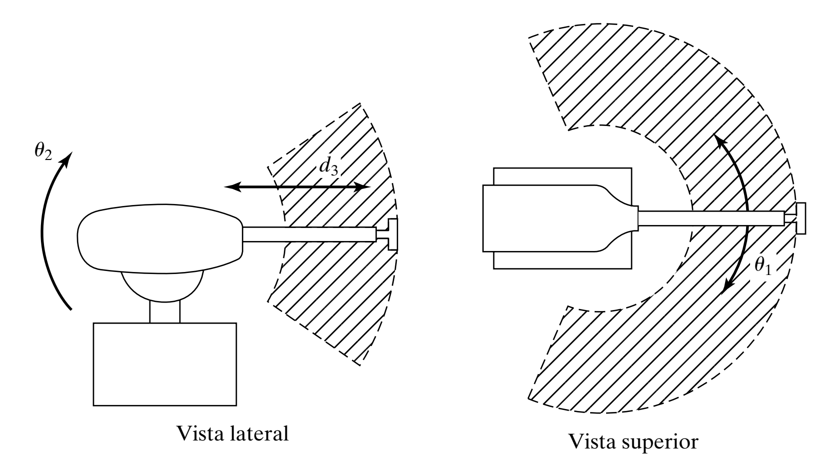


FIGURA 8.6: Un manipulador esférico.

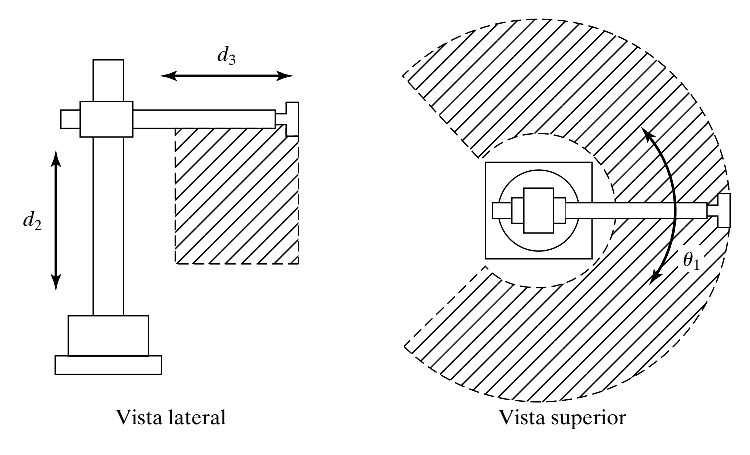


FIGURA 8.7: Un manipulador cilíndrico.

Muñecas

Las configuraciones de muñeca más comunes consisten en dos o tres articulaciones angulares con ejes ortogonales que se intersecan. Por lo regular, la primera articulación de muñeca forma la articulación 4 del manipulador.

Una configuración de tres ejes ortogonales garantiza que pueda lograrse cualquier orientación (suponiendo que no haya límites de los ángulos de las articulaciones) [2]. Como se mencionó en el capítulo 4, cualquier manipulador con tres ejes consecutivos que se intersecan poseerá una solución cinemática de forma cerrada. Por lo tanto, una muñeca con tres ejes ortogonales puede ubicarse en el extremo del manipulador en cualquier orientación deseada sin problemas. La figura 8.8 es un diagrama esquemático de un posible diseño de dicha muñeca que utiliza varios conjuntos de engranajes cónicos para controlar el mecanismo desde actuadores remotamente ubicados.

En la práctica, es difícil construir una muñeca con tres ejes ortogonales que no esté sujeta a limitaciones bastante severas de los ángulos de las articulaciones. Un interesante

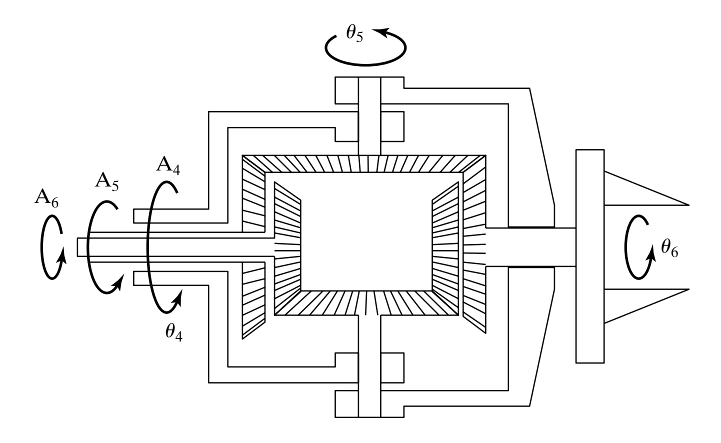


FIGURA 8.8: Una muñeca con eje ortogonal controlada por actuadores remotamente ubicados, mediante tres ejes concéntricos.

diseño utilizado en varios robots fabricados por Cincinatti Milacron (figura 1.4) emplea una muñeca con tres ejes que se intersecan pero que no son ortogonales. En este diseño (llamado “muñeca de triple balanceo”), las tres articulaciones de la muñeca pueden girar continuamente sin límites. Sin embargo, el hecho de que los ejes no sean ortogonales crea un conjunto de orientaciones que son imposibles de alcanzar con esta muñeca. Este conjunto de orientaciones inalcanzables se describe mediante un cono dentro del cual no puede estar el tercer eje de la muñeca. (Vea el ejercicio 8.11.) Sin embargo, la muñeca puede montarse en el vínculo 3 del manipulador de tal forma que la estructura de los vínculos ocupe este cono, por lo que se bloquearía el acceso de todas formas. La figura 8.9 muestra dos dibujos de dicha muñeca [24].

Algunos robots industriales tienen muñecas que *no* tienen ejes que se intersequen. Esto implica que tal vez no exista una solución cinemática de forma cerrada. No obstante, si la muñeca está montada en un manipulador articulado de tal forma que el eje de la articulación 4 sea paralelo a los ejes de las articulaciones 2 y 3, como en la figura 8.10, sí *habrá* una solución cinemática de forma cerrada. De igual forma, una muñeca con ejes que no se intersequen, que esté montada en un robot cartesiano, permite que el manipulador pueda resolverse de forma cerrada.

Generalmente, los robots 5-DOF para soldar utilizan muñecas con dos ejes orientadas como se muestra en la figura 8.11. Observe que si el robot tiene una herramienta simétrica, esta “articulación ficticia” debe seguir las reglas del diseño de las muñecas. Esto significa que, para llegar a todas las orientaciones, la herramienta debe estar montada con su eje de simetría ortogonal al eje de la articulación 5. En el peor de los casos, cuando el eje de simetría es paralelo al eje de la articulación 5, el sexto eje ficticio se encuentra en una configuración permanentemente singular.

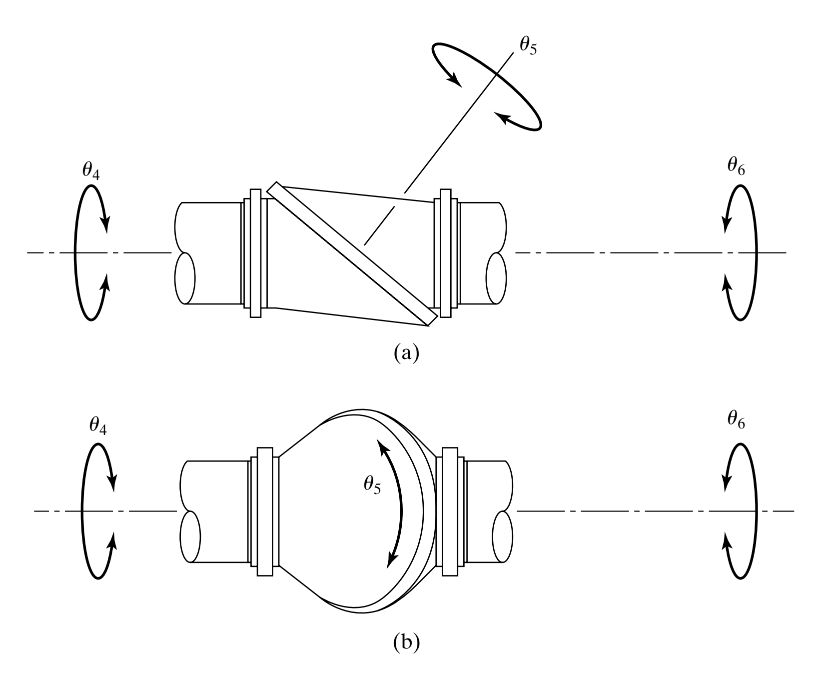


FIGURA 8.9: Dos vistas de una muñeca con eje no ortogonal. De *International Encyclopedia of Robotics*, por R. Dorf y S. Nof (editores). De “Wrists” por M. Rosheim, John C. Wiley and Sons, Inc., NuevaYork, NY ©1988. Reimpreso con permiso.

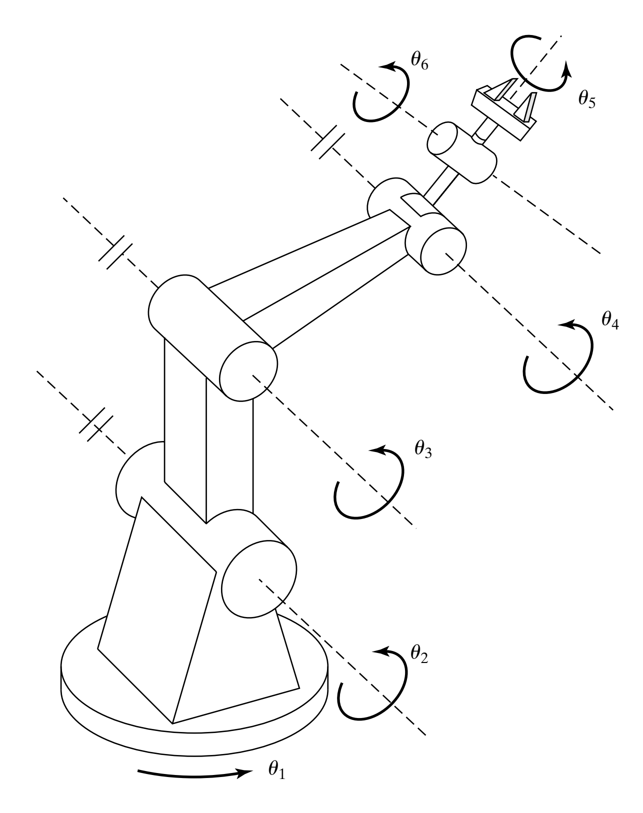


FIGURA 8.10: Un manipulador con una muñeca cuyos ejes no se intersecan. Sin embargo, este robot posee una solución cinemática de forma cerrada.

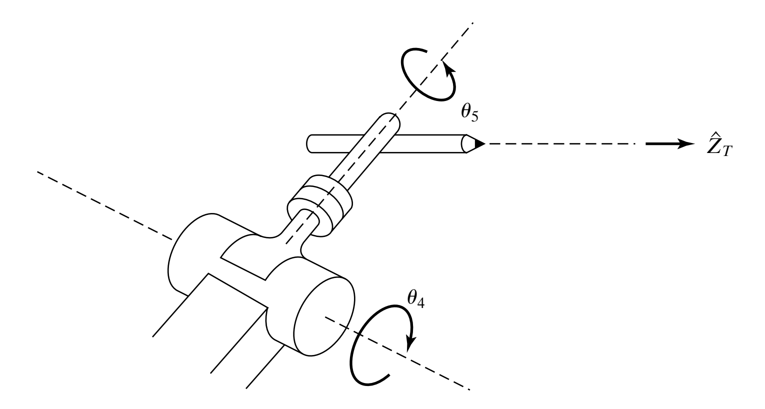


FIGURA 8.11: Diseño típico de muñeca de un robot 5-DOF para soldar.

8.4 MEDIDAS CUANTITATIVAS DE ATRIBUTOS DEL ESPACIO DE TRABAJO

Los diseñadores de manipuladores han propuesto varias medidas cuantitativas interesantes de varios atributos de los espacios de trabajo.

Eficiencia del diseño en términos de la generación del espacio de trabajo

Algunos diseñadores se dieron cuenta de que parecía necesitarse más material para construir un manipulador cartesiano que para construir un manipulador articulado con un volumen similar de espacio de trabajo. Para darnos una idea cuantitativa, primero definiremos la **suma de longitud** de un manipulador, como

$$L = \sum_{i=1}^{N}(a_{i-1} + d_i), \tag{8.1}$$

en donde $a_{i\text{-}1}$ y d_i son la longitud de vínculo y el desplazamiento de vínculo, según se definieron en el capítulo 3. En consecuencia, la suma de longitud de un manipulador nos da una medida aproximada de la “longitud” de todos los vínculos. Observe que para las articulaciones prismáticas, d_i debe interpretarse aquí como una constante igual a la longitud de recorrido entre los límites del recorrido de las articulaciones.

En [3] el **índice de longitud estructural** Q_L se define como el cociente de la suma de longitud del manipulador y la raíz cúbica del volumen del espacio de trabajo; esto es,

$$Q_L = L/\sqrt[3]{w}, \tag{8.2}$$

en donde L se da en la ecuación (8.1) y W es el volumen del espacio de trabajo del manipulador. Por lo tanto, Q_L intenta indizar la cantidad relativa de estructura (longitud de vínculos) requerida por distintas configuraciones para generar un volumen de trabajo dado. En consecuencia, un buen diseño sería aquel en el que un manipulador con una pequeña suma de longitud tuviera, sin embargo, un volumen grande de espacio de trabajo. Los buenos diseños tienen una Q_L baja.

Si consideramos sólo la estructura de colocación de un manipulador cartesiano (y por lo tanto el espacio de trabajo del punto de la muñeca), el valor de Q_L se minimiza cuando las tres articulaciones tienen la misma longitud de recorrido. Este valor mínimo es $Q_L = 3.0$. Por otra parte, un manipulador articulado ideal como el de la figura 8.4 tiene $Q_L = \frac{1}{\sqrt[3]{4\pi/3}} \cong 0.62$. Esto ayuda a cuantificar nuestra anterior declaración de que los manipuladores articulados son superiores a otras configuraciones, ya que tienen una mínima intrusión en su propio espacio de trabajo. Desde luego, que en cualquier estructura de un manipulador real, el número que acabamos de dar se haría algo más grande por el efecto de los límites de articulación, al reducir el volumen del espacio de trabajo.

EJEMPLO 8.1

Un manipulador SCARA, como el de la figura 8.5, tiene los vínculos 1 y 2 de igual longitud, representada por $l/2$, y el rango de movimiento de la articulación prismática 3 está dado por d_3. Suponga por simplicidad que los límites de las articulaciones están ausentes, y encuentre Q_L. ¿Qué valor de d_3 minimiza Q_L y cuál es este valor mínimo?

La suma de longitud de este manipulador es $L = l/2 + l/2 + d_3 = l + d_3$, y el volumen del espacio de trabajo es el de un cilindro recto de radio l y altura d_3; por lo tanto,

$$Q_L = \frac{l + d_3}{\sqrt[3]{\pi l^2 d_3}}. \tag{8.3}$$

Si minimizamos Q_L como una función de la proporción d_3/l obtenemos $d_3 = l/2$ como valor óptimo [3]. El valor mínimo correspondiente de Q_L sería 1.29.

Diseño de espacios de trabajo bien condicionados

En puntos singulares, un manipulador pierde efectivamente uno o más grados de libertad, por lo que tal vez no pueda realizar ciertas tareas en esos puntos. De hecho, en las proximidades de los puntos singulares (incluyendo las singularidades en los límites del espacio de trabajo), las acciones del manipulador podrían dejar de estar **bien condicionadas**. En cierto sentido, entre más alejado esté el manipulador de las singularidades podrá moverse mejor de manera uniforme y podrá aplicar fuerzas uniformemente en todas las direcciones. Se han sugerido varias medidas para cuantificar este efecto. El uso de dichas medidas en tiempo de diseño podría producir un diseño de manipulador con un subespacio bien condicionado del espacio de trabajo con el mayor tamaño posible.

Las configuraciones singulares se dan mediante

$$\det(J(\Theta)) = 0, \tag{8.4}$$

por lo tanto, es natural utilizar el determinante del jacobiano en una medida de la destreza del manipulador. En [4], la **medida de manipulabilidad** w se define como

$$w = \sqrt{\det(J(\Theta)J^T(\Theta))}, \tag{8.5}$$

que para un manipulador no redundante se reduce a

$$w = |\det(J(\Theta))|. \tag{8.6}$$

Un buen diseño de manipulador tiene grandes áreas de su espacio de trabajo caracterizadas por altos valores de w.

Mientras que el análisis de velocidad motivó la ecuación (8.6), otros investigadores han propuesto medidas de manipulabilidad basadas en el análisis de aceleración o en la capacidad de aplicación de fuerza. Asada [5] sugirió un examen de los valores propios de la matriz de masas cartesiana

$$M_x(\Theta) = J^{-T}(\Theta)M(\Theta)J^{-1}(\Theta) \tag{8.7}$$

como una medida de qué tan bien puede acelerar el manipulador en varias direcciones cartesianas; él sugiere una representación gráfica de esta medida como un **elipsoide de inercia** dado por

$$X^T M_x(\Theta)X = 1, \tag{8.8}$$

que es la ecuación de una elipse n-dimensional, en donde n es la dimensión de X. Los ejes del elipsoide dados en la ecuación (8.8) yacen en las direcciones de los vectores propios de $M_X(\Theta)$, y los recíprocos de las raíces cuadradas de los valores propios correspondientes proveen las longitudes de los ejes en el elipsoide. Los puntos bien condicionados en el espacio de trabajo del manipulador se caracterizan por elipsoides de inercia que son esféricos (o que casi lo son).

La figura 8.12 muestra gráficamente las propiedades de un manipulador planar de dos vínculos. En el centro del espacio de trabajo, el manipulador está bien condicionado, como se indica mediante elipsoides casi circulares. En los límites del espacio de trabajo las elipses se aplanan, indicando la dificultad del manipulador para acelerar en ciertas direcciones.

En [6–8, 25] se han propuesto otras medidas de condicionamiento del espacio de trabajo.

8.5 ESTRUCTURAS REDUNDANTES Y DE CADENA CERRADA

El alcance de este libro está limitado a los manipuladores que están formados por conjuntos de vínculos de cadena serial de seis o menos articulaciones. No obstante, en esta sección hablaremos brevemente sobre los manipuladores que no pertenecen a esta clase.

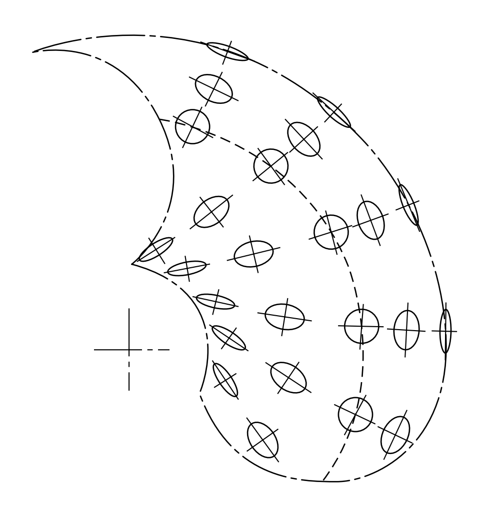


FIGURA 8.12: El espacio de trabajo de un brazo planar 2-DOF, mostrando los elipsoides de inercia, de [5] (©1984 IEEE). La línea punteada indica un lugar de puntos isotrópicos en el espacio de trabajo. Reimpresa con permiso.

Micromanipuladores y otras redundancias

La capacidad de posicionamiento espacial general requiere solamente de seis grados de libertad, pero existen ventajas al tener articulaciones aún más controlables.

Uno de los usos para estas libertades adicionales ya tiene cierta aplicación práctica [9, 10], además de que es cada vez más interesante para la comunidad de investigación: un **micromanipulador**. Generalmente un micromanipulador está formado por varios grados, rápidos y precisos, de libertad ubicados cerca del extremo distal de un manipulador "convencional". Este manipulador convencional se encarga de los movimientos extensos, mientras que el micromanipulador, cuyas articulaciones por lo general tienen un rango de movimiento más pequeño, realiza el movimiento fino y el control de la fuerza.

Como se sugiere en [11, 12], las articulaciones adicionales también pueden ayudar a que un mecanismo evite las configuraciones singulares, por ejemplo, cualquier muñeca con tres grados de libertad sufrirá de configuraciones singulares (cuando los tres ejes estén sobre un plano), pero una muñeca con cuatro grados de libertad puede evitar efectivamente dichas configuraciones [13-15].

La figura 8.13 muestra dos configuraciones sugeridas [11, 12] para los manipuladores con siete grados de libertad.

Un uso importante de los robots redundantes es para evitar colisiones mientras se opera en entornos de trabajo llenos de obstáculos. Como hemos visto, un manipulador con seis grados de libertad puede llegar a una posición y orientación dadas en un número determinado de maneras. La adición de una séptima articulación permite un número infinito de formas, lo que hace que la necesidad de evitar obstáculos influya en la decisión.

Estructuras de lazo cerrado

Aunque hemos considerado sólo manipuladores de cadena serial en nuestro análisis, algunos manipuladores contienen **estructuras de lazo cerrado**. Por ejemplo, el robot Motoman L-3 descrito en los capítulos 3 y 4 posee estructuras de lazo cerrado en el mecanismo de control de las articulaciones 2 y 3. Las estructuras de lazo cerrado ofrecen un beneficio: un

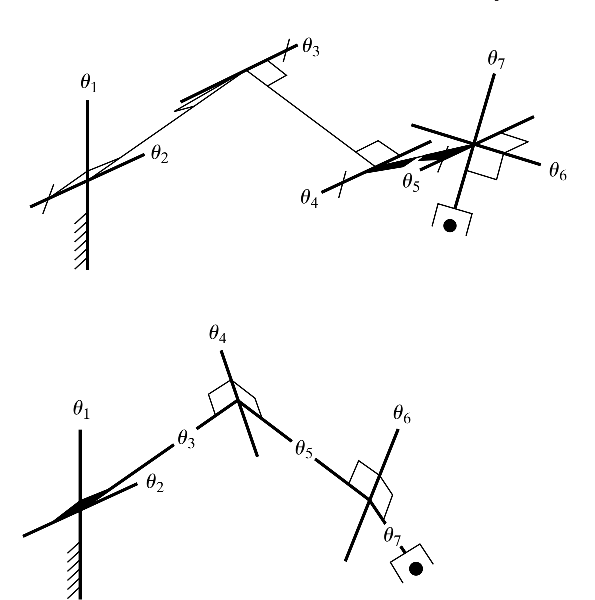


FIGURA 8.13: Dos diseños sugeridos de manipulador con siete grados de libertad [3].

aumento en la rigidez del mecanismo [16]. Por otro lado, las estructuras de lazo cerrado generalmente reducen el rango permisible de movimiento de las articulaciones y, por ende, reducen el tamaño del espacio de trabajo.

En la figura 8.14 se muestra un **mecanismo Stewart**, una alternativa de lazo cerrado para el manipulador 6-DOF serial. La posición y la orientación del "efector final" se controlan mediante las longitudes de los seis actuadores lineales que lo conectan a la base, en cuyo extremo cada actuador está conectado por una articulación universal con dos grados de libertad. En el efector final, cada actuador está unido con una articulación cotiloidea (es decir, de "bola y ranura") de tres grados de libertad. Exhibe características comunes a la mayoría de los mecanismos de lazo cerrado: puede hacerse muy rígido, pero los vínculos tienen un rango de movimiento mucho más limitado que los vínculos seriales. Específicamente, el mecanismo Stewart demuestra una reversión interesante en la naturaleza de las soluciones cinemáticas directa e inversa: la solución inversa es considerablemente simple, mientras que la solución directa es por lo común bastante compleja, y algunas veces carece de una formulación de forma cerrada. (Vea los ejercicios 8.7 y 8.12).

En general, el número de grados de libertad de un mecanismo de lazo cerrado no es obvio y puede calcularse mediante la fórmula de **Grübler** [17],

$$F = 6(l - n - 1) + \sum_{i=1}^{n} f_i, \tag{8.9}$$

en donde F es el número total de grados de libertad en el mecanismo, l es el número de vínculos (incluyendo la base), n es el número total de articulaciones y f_i es el número de grados de libertad asociados con la i-ésima articulación. Una versión en dos dimensiones de la fórmula de Grübler (cuando se considera que todos los objetos tienen tres grados de libertad si no hay restricciones) se obtiene sustituyendo el 6 en la ecuación (8.9) con un 3.

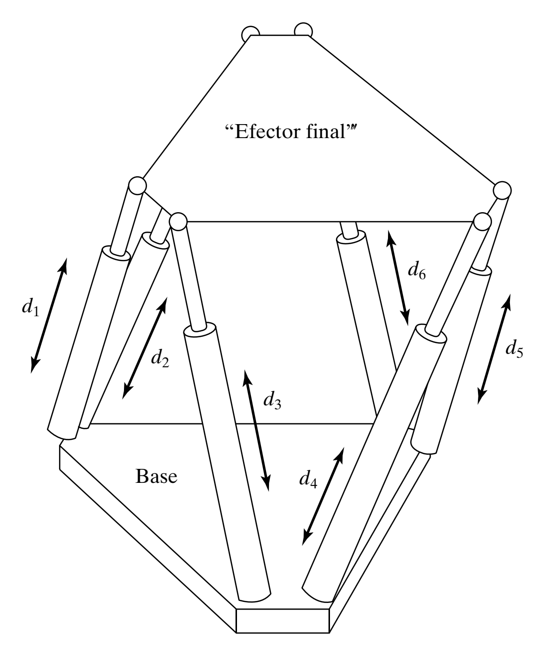


FIGURA 8.14: El mecanismo Stewart es un manipulador de seis grados de libertad completamente paralelo.

EJEMPLO 8.2

Utilice la fórmula de Grübler para verificar que el mecanismo Stewart (figura 8.14) realmente tenga seis grados de libertad.

El número de articulaciones es de 18 (6 universales, 6 cotiloideas y 6 prismáticas en los actuadores). El número de vínculos es 14 (2 piezas para cada actuador, el efector final y la base). La suma de todos los grados de libertad de las articulaciones es de 36. Utilizando la fórmula de Grübler, podemos verificar que el número total de grados de libertad es de seis:

$$F = 6(14 - 18 - 1) + 36 = 6. \tag{8.10}$$

8.6 ESQUEMAS DE ACCIONAMIENTO

Una vez elegida la estructura cinemática general de un manipulador, el siguiente aspecto más importante es el accionamiento de las articulaciones. Por lo general, el actuador, la reducción y la transmisión están estrechamente acoplados y deben diseñarse en conjunto.

Ubicación del actuador

La elección más simple para la ubicación del actuador es en la articulación que controla o cerca de ella. Si el actuador puede producir suficiente momento de torsión o fuerza, su salida puede unirse directamente a la articulación. Este arreglo, conocido como configuración de **control directo** [18], ofrece las ventajas de simplicidad en el diseño y capacidad de control superior; esto es, sin elementos de transmisión o de reducción entre el actuador y la articulación, los movimientos de la articulación pueden controlarse con la misma fidelidad que el mismo actuador.

Por desgracia, muchos actuadores están mejor adaptados a velocidades relativamente altas y momentos de torsión bajos, por lo cual requieren de un **sistema de reducción de velocidad**. Además, los actuadores tienden a ser demasiado pesados. Si pueden ubicarse apartados de la articulación y hacia la base del manipulador, la inercia total de éste podrá reducirse considerablemente. Esto a su vez reduce el tamaño necesario para los actuadores. Para obtener estos beneficios se necesita un **sistema de transmisión** para transferir el movimiento del actuador a la articulación.

En un sistema de control de articulaciones con un actuador montado en forma remota, el sistema de reducción podría colocarse tanto en el actuador como en la articulación. Algunos arreglos combinan las funciones de transmisión y reducción. Aparte del aumento en la complejidad, la principal desventaja de los sistemas de reducción y transmisión es que introducen más fricción y flexibilidad en el mecanismo. Cuando la reducción es en la articulación, la transmisión estará trabajando a velocidades mayores y momentos de torsión más bajos. Un momento de torsión más bajo significa que la flexibilidad será un problema menos. No obstante, si el peso del reductor es considerable, se perderá algo de la ventaja de tener los actuadores montados en forma remota.

En el capítulo 3 se dieron los detalles para el esquema de accionamiento del Yasukawa Motoman L-3, que tiene el típico diseño en el que los actuadores están montados en forma remota y los movimientos resultantes de las articulaciones están acoplados. Las ecuaciones (3.16) muestran explícitamente cómo los movimientos del actuador pueden causar movimientos de la articulación. Por ejemplo, observe que el movimiento del actuador 2 produce el movimiento de las articulaciones 2, 3 y 4.

La distribución óptima de las etapas de reducción a lo largo de la transmisión dependerá en última instancia de la flexibilidad de la transmisión, del peso del sistema de reducción, de la fricción asociada con el sistema de reducción y de la facilidad para incorporar estos componentes en el diseño del manipulador en general.

Sistemas de reducción y de transmisión

Los **engranajes** son el elemento más común utilizado para la reducción. Pueden producir grandes reducciones en configuraciones relativamente compactas. Los pares de engranajes vienen en varias configuraciones para ejes paralelos (engranajes rectos), ejes ortogonales que se intersecan (engranajes cónicos), ejes oblicuos (engranajes de tornillos sinfín o engranajes helicoidales cruzados), entre otras. Los distintos tipos de engranajes tienen diferentes clasificaciones de carga, características de desgaste y propiedades friccionales.

Las principales desventajas de usar engranajes son un aumento en el **efecto retroactivo** y en la fricción. El efecto retroactivo, que se produce debido a que los engranajes no encajan correctamente, puede definirse como el máximo movimiento angular de los engranajes de salida cuando el engranaje de entrada permanece fijo. Si los dientes del engranaje encajan firmemente para eliminar el efecto retroactivo, puede haber cantidades excesivas de fricción. Un engranaje y montaje muy precisos minimizan estos problemas, pero también aumentan el costo.

La **proporción del engranaje** η describe los efectos de reducción de velocidad y aumento en el momento de torsión de un par de engranajes. Para los sistemas de reducción de velocidad definiremos $\eta > 1$; por lo que las relaciones entre las velocidades y los momentos de torsión de entrada y salida se dan mediante

$$\begin{aligned} \dot{\theta}_o &= (1/\eta)\dot{\theta}_i \\ \tau_o &= \eta\tau_i, \end{aligned} \tag{8.11}$$

en donde $\dot{\theta}_o$ y $\dot{\theta}_i$ son las velocidades de salida y de entrada, respectivamente, y τ_o y τ_i los momentos de torsión de salida y entrada correspondientes.

La segunda clase extensa de elementos de reducción incluye bandas flexibles, cables y correas. Como todos estos elementos deben ser lo suficientemente flexibles como para doblarse alrededor de las poleas, también tienden a ser flexibles en la dirección longitudinal. La flexibilidad de estos elementos es proporcional a su longitud, y dado que estos sistemas son flexibles, debe haber algún mecanismo para cargar previamente el lazo y asegurarse de que la banda o el cable permanezcan enganchados en la polea. Las cargas previas grandes pueden agregar una tensión indebida al elemento flexible e introducir una fricción excesiva.

Los cables o bandas flexibles pueden usarse en un lazo cerrado o como elementos de un solo extremo que siempre se mantienen en tensión mediante cierto tipo de carga previa. En una articulación que es accionada por resorte en una dirección podría usarse un cable de un solo extremo para jalarla. Como alternativa, dos sistemas activos de un solo extremo pueden oponerse uno al otro. Este arreglo elimina el problema de cargas previas excesivas, pero agrega más actuadores.

Las cadenas de rodillos son similares a las bandas flexibles, pero pueden doblarse alrededor de poleas relativamente pequeñas al mismo tiempo que conservan una alta rigidez. Como resultado del desgaste y de cargas pesadas en los pernos que conectan los vínculos, los sistemas con bandas dentadas son más compactos que las cadenas de rodillos para ciertas aplicaciones.

Las transmisiones de banda, cable, correa y cadena tienen la habilidad de combinar la transmisión con la reducción. Como se muestra en la figura 8.15, cuando la polea de entrada tiene un radio r_1 y la de salida un radio r_2, la proporción de "engranaje" del sistema de transmisión es

$$\eta = \frac{r_2}{r_1}. \tag{8.12}$$

Los tornillos de posicionamiento, o los de cojinete tipo bola son otro método popular para obtener una gran reducción en un paquete compacto (figura 8.16). Los tornillos de posicionamiento son muy rígidos y pueden soportar cargas muy grandes; además tienen la propiedad de que transforman el movimiento giratorio en movimiento lineal. Los tornillos de cojinete tipo bola son similares a los de posicionamiento, pero en vez de que las roscas de las tuercas se monten directamente en las roscas de los tornillos, un circuito recirculante de cojinetes tipo bola rueda entre los conjuntos de roscas. Los tornillos de cojinete tipo bola tienen muy baja fricción y por lo general cuentan con arranque trasero.

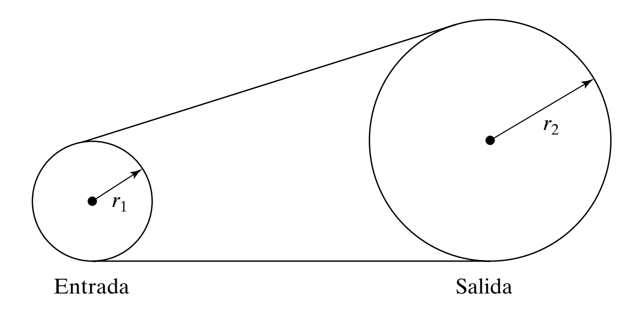


FIGURA 8.15: Las transmisiones de banda, cable, correa y cadena tienen la habilidad de combinar la transmisión con la reducción.

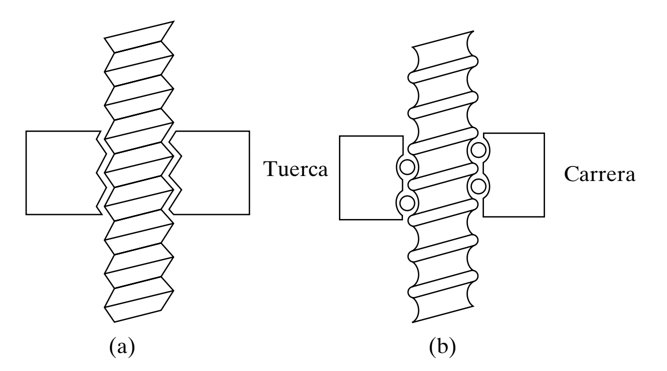


FIGURA 8.16: Los tornillos de posicionamiento (a) y los tornillos de cojinete tipo bola (b) combinan una gran reducción con la transformación de movimiento giratorio en movimiento lineal.

8.7 RIGIDEZ Y DEFLEXIONES

Un objetivo importante en el diseño de la mayoría de los manipuladores es la rigidez total de la estructura y el sistema de control. Los sistemas rígidos tienen dos beneficios importantes: En primer lugar, como los manipuladores comunes no tienen sensores para medir directamente la ubicación de la trama herramienta, ésta se calcula utilizando la cinemática directa con base en las posiciones detectadas de las articulaciones. Para un cálculo preciso, los vínculos no pueden pandearse por la gravedad u otras cargas. En otras palabras, deseamos que nuestra descripción Denavit-Hartenberg de los vínculos permanezca fija bajo varias condiciones de carga. En segundo lugar, las flexibilidades en la estructura o el tren de transmisión producirán **resonancias**, las cuales tienen un efecto indeseable en el desempeño de un manipulador. En esta sección consideraremos las cuestiones de rigidez y las deflexiones resultantes bajo cargas. Pospondremos para el capítulo 9 una discusión más detallada de las resonancias.

Elementos flexibles en paralelo y en serie

Como puede mostrarse fácilmente (vea el ejercicio 8.21), la combinación de dos miembros flexibles de rigidez k_1 y k_2 “conectados en paralelo” produce la siguiente rigidez neta:

$$k_{\text{paralelo}} = k_1 + k_2; \tag{8.13}$$

“conectados en serie”, la combinación produce la siguiente rigidez neta:

$$\frac{1}{k_{\text{serie}}} = \frac{1}{k_1} + \frac{1}{k_2}. \tag{8.14}$$

Al considerar los sistemas de transmisión, a menudo tenemos el caso de una etapa de reducción o de transmisión en serie con una etapa subsiguiente de reducción o de transmisión; por lo tanto la ecuación (8.14) se vuelve útil.

Ejes

Un método común para transmitir el movimiento giratorio es a través de ejes. La rigidez torsional de un eje redondo puede calcularse [19] así:

$$k = \frac{G\pi d^4}{32l}, \tag{8.15}$$

en donde d es el diámetro del eje, l es su longitud y G es el módulo de elasticidad transversal (aproximadamente 7.5×10^{10} Nt/m^2 para el acero, y un tercio de esa cantidad para el aluminio).

Engranajes

Aunque por lo general los engranajes son bastante rígidos, agregan elasticidad en el sistema de accionamiento. Una fórmula aproximada para estimar la rigidez del engranaje de salida (suponiendo que el engranaje de entrada sea fijo) se da en [20] como

$$k = C_g b r^2, \tag{8.16}$$

en donde b es la anchura lateral de los engranajes, r es el radio del engranaje de salida y $C_g = 1.34 \times 10^{10}$ Nt/m^2 para el acero.

Los engranajes también tienen el efecto de cambiar la rigidez efectiva del sistema de control por un factor de η^2. Si la rigidez del sistema de transmisión antes de la reducción (es decir, en el lado de entrada) es k_i, de manera que

$$\tau_i = k_i \delta\theta_i, \tag{8.17}$$

y la rigidez del lado de salida de la reducción es k_o, de manera que

$$\tau_o = k_o \delta\theta_o, \tag{8.18}$$

entonces podemos calcular la relación entre k_i y k_o (bajo la suposición de un par de engranajes perfectamente rígidos) como

$$k_o = \frac{\tau_o}{\delta\theta_o} = \frac{\eta k_i \delta\theta_i}{(1/\eta)\delta\theta_i} = \eta^2 k_i. \tag{8.19}$$

Por lo tanto, una reducción de engranajes tiene el efecto de aumentar la rigidez según el cuadrado de la proporción de los engranajes.

EJEMPLO 8.3

Un eje con una rigidez torsional igual a 500.0 Nt-m/radián está conectado del lado de entrada de un conjunto de engranajes con $\eta = 10$, cuyo engranaje de salida (cuando el de entrada es fijo) exhibe una rigidez de 5000.0 Nt-m/radián. ¿Cuál es la rigidez de salida del sistema de accionamiento combinado?

Con las ecuaciones (8.14) y (8.19) tenemos que

$$\frac{1}{k_{serie}} = \frac{1}{5000.0} + \frac{1}{10^2(500.0)}, \tag{8.20}$$

o

$$k_{serie} = \frac{50000}{11} \cong 4545.4 \text{ Nt m/radian} \tag{8.21}$$

Cuando una reducción de velocidad relativamente grande es el último elemento de un sistema de transmisión con múltiples elementos, la rigidez de los elementos anteriores por lo general puede ignorarse.

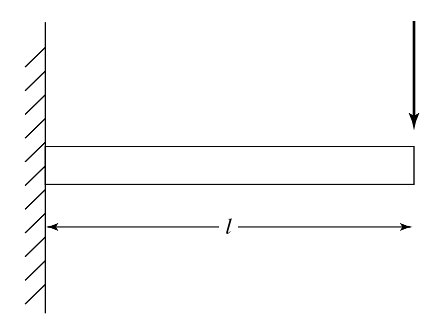


FIGURA 8.17: Una simple viga voladiza utilizada para modelar la rigidez de un vínculo a una carga en el extremo.

Correas

En una transmisión de correa tal como la de la figura 8.15, la rigidez se obtiene mediante

$$k = \frac{AE}{l}, \tag{8.22}$$

en donde A es el área de corte transversal de la correa, E es el módulo de elasticidad y l la longitud libre entre las poleas, más un tercio de la longitud de la correa que está en contacto con las poleas [19].

Vínculos

Como una aproximación de la rigidez de un vínculo, podríamos modelar un vínculo individual como una viga voladiza y calcular la rigidez en el punto extremo, como en la figura 8.17. Para una viga hueca redonda, esta rigidez se obtiene mediante [19]

$$k = \frac{3\pi E(d_o^4 - d_i^4)}{64l^3}, \tag{8.23}$$

en donde d_i y d_o son los diámetros interno y externo de la viga tubular, l es longitud y E el módulo de elasticidad (aproximadamente 2×10^{11} Nt/m^2 para el acero y un tercio de esa cantidad para el aluminio). Para una viga hueca de corte cuadrado transversal, esta rigidez se obtiene mediante

$$k = \frac{E(w_o^4 - w_i^4)}{4l^3}, \tag{8.24}$$

en donde w_i y w_o son las anchuras interior y exterior de la viga (es decir, el grosor de pared es $w_o - w_i$).

EJEMPLO 8.4

Un vínculo de corte transversal cuadrado de $5 \times 5 \times 50$ cm, con un grosor de pared de 1 cm, es controlado por un conjunto de engranajes rígidos con $\eta = 10$, y la entrada de los engranajes está controlada por un eje con un diámetro de 0.5 cm y una longitud de 30 cm. ¿Cuál es la deflexión ocasionada por una fuerza de 100 Nt en el extremo del vínculo?

Si utilizamos la ecuación (8.24) podemos calcular la rigidez del vínculo como

$$k_{\text{vínculo}} = \frac{(2 \times 10^{11})(0.05^4 - 0.04^4)}{4(0.5)} \cong 3.69 \times 10^5. \tag{8.25}$$

En consecuencia, para una carga de 100 Nt hay una deflexión en el mismo vínculo de

$$\delta x = \frac{100}{k_{\text{vínculo}}} \cong 2.7 \times 10^{-4} \text{ m}, \tag{8.26}$$

o 0.027 cm.

Además, los 100 Nt en el extremo de un vínculo de 50 cm están aplicando un momento de torsión de 50 Nt-m en el engranaje de salida. Los engranajes son rígidos, pero la flexibilidad del eje de entrada es

$$k_{\text{eje}} = \frac{(7.5 \times 10^{10})(3.14)(5 \times 10^{-3})^4}{(32)(0.3)} \cong 15.3 \text{ Nt m/radian} \tag{8.27}$$

lo cual, visto desde el engranaje de salida es

$$k'_{\text{eje}} = (15.3)(10^2) = 1530.0 \text{ Nt-m/radian} \tag{8.28}$$

Si se carga con 50 Nt-m se produce una deflexión angular de

$$\delta\theta = \frac{50.0}{1530.0} \cong 0.0326 \text{ radian} \tag{8.29}$$

por lo que la deflexión lineal total en la punta del vínculo sería

$$\delta x \cong 0.027 + (0.0326)(50) = 0.027 + 1.630 = 1.657 \text{ cm}. \tag{8.30}$$

En nuestra solución hemos supuesto que el eje y el vínculo están hechos de acero. La rigidez de ambos miembros es lineal en E, el módulo de elasticidad, por lo que para los elementos de aluminio podemos multiplicar nuestro resultado por 3 (aproximadamente).

En esta sección hemos examinado ciertas fórmulas simples para estimar la rigidez de los engranajes, ejes, correas y vínculos. El propósito de estas fórmulas es obtener ciertos lineamientos para definir el tamaño de los miembros estructurales y los elementos de transmisión. No obstante, en aplicaciones prácticas muchas fuentes de flexibilidad son muy difíciles de modelar. A menudo el tren de transmisión introduce una flexibilidad considerablemente mayor que el vínculo de un manipulador. Además, aquí no hemos considerado muchas fuentes de flexibilidad (de los cojinetes, del montaje del actuador, etc.) en el sistema de control. Generalmente cualquier intento de predecir la rigidez en forma analítica genera una predicción excesivamente alta, ya que no se consideran muchas fuentes.

Pueden utilizarse **técnicas de elementos finitos** para predecir con más precisión la rigidez (y otras propiedades) de elementos estructurales más realistas. Éste es todo un campo por sí solo [21] y está más allá del alcance de este libro.

Actuadores

De entre varios actuadores, los **cilindros hidráulicos** o **actuadores de álabe** eran originalmente los más populares para utilizarse en los manipuladores. En un paquete

relativamente compacto pueden producir suficiente fuerza para controlar las articulaciones sin un sistema de reducción. La velocidad de operación depende del sistema de bomba y acumulador, que por lo general se ubica lejos del manipulador. El control de posición de un sistema hidráulico es bien conocido y relativamente simple. Los primeros robots industriales y muchos robots industriales modernos de gran tamaño utilizaron o utilizan actuadores hidráulicos.

Desafortunadamente la hidráulica requiere una gran cantidad de equipo, como bombas, acumuladores, mangueras y servo válvulas. Los sistemas hidráulicos también tienden a ser inherentemente desordenados, lo que los hace inadecuados para ciertas aplicaciones. Con la llegada de estrategias de control de robots más avanzadas, en las que las fuerzas de los actuadores deben aplicarse con precisión, la hidráulica está en desventaja debido a la fricción contribuida por sus obturaciones.

Los **cilindros neumáticos** poseen todos los atributos favorables de la hidráulica, además de que son más limpios (expelen aire en vez de fluido hidráulico). No obstante, los actuadores neumáticos son difíciles de controlar con precisión debido a la compresibilidad del aire y la alta fricción de las obturaciones.

Los motores eléctricos son el tipo de actuador más popular para los manipuladores y, aunque no tienen la proporción de energía-peso de sus contrapartes hidráulicas o neumáticas, su capacidad de control y fácil interconexión los hace atractivos para manipuladores de tamaño pequeño o mediano.

Los motores con escobillas de corriente directa (CD) (figura 8.18) son los más simples de coordinar y controlar. La corriente se conduce hacia los devanados del rotor mediante escobillas, las cuales hacen contacto con el conmutador giratorio; pero el desgaste y la fricción de éstas puede ocasionar problemas. Los nuevos materiales magnéticos han hecho posible obtener momentos de torsión máximos elevados. El factor limitante en la producción del momento de torsión de estos motores es el sobrecalentamiento de los

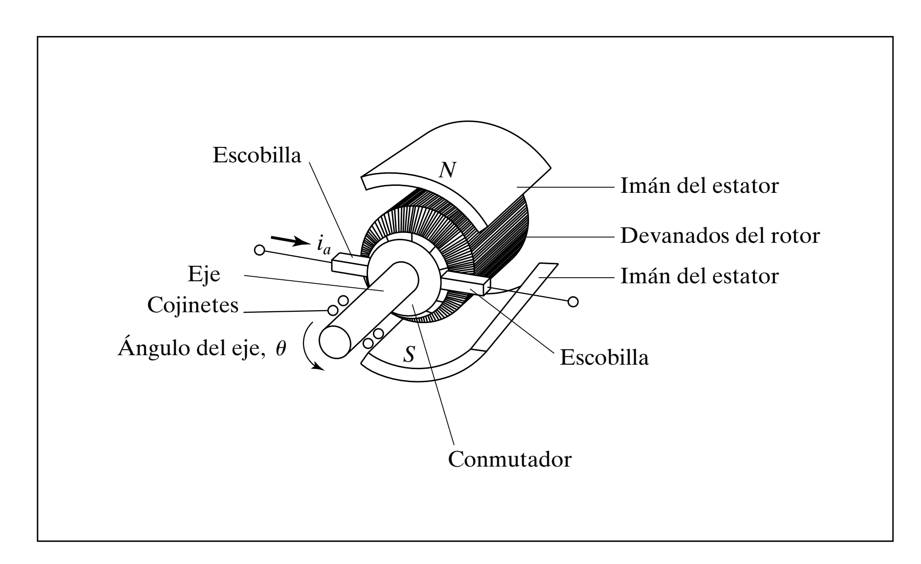


FIGURA 8.18: Los motores de escobillas de CD están entre los actuadores que se utilizan con más frecuencia en el diseño de manipuladores. Franklin, Powell, Enami-Naeini, *Feedback Control of Dynamic Systems*, ©1988, Addison-Wesley, Reading, MA. Reimpreso con permiso.

devanados. En ciclos cortos pueden lograrse momentos de torsión elevados, pero durante periodos largos sólo pueden sostenerse momentos de torsión mucho menores.

Los **motores sin escobillas** solucionan los problemas de desgaste y fricción de las escobillas. Aquí los devanados permanecen fijos y la pieza del campo magnético gira. Un sensor en el rotor detecta el ángulo del eje y es utilizado por los componentes electrónicos externos para realizar la conmutación. Otra ventaja de los motores sin escobillas es que el devanado está en el exterior, unido a la cubierta del motor, lo que permite un mejor enfriamiento. Los valores sostenidos del momento de torsión tienden a ser algo más altos que para los motores con escobillas de tamaño similar.

Los motores de corriente alterna (CA) y los motores de pasos se han utilizado con poca frecuencia en la robótica industrial. La dificultad de control de los primeros y la baja capacidad para generar momento de torsión de los segundos han limitado su uso.

8.8 DETECCIÓN DE POSICIÓN

Casi todos los manipuladores son mecanismos servo-controlados; esto es, el comando de fuerza o de momento de torsión para un actuador está basado en el error entre la posición detectada de la articulación y la posición deseada. Para ello se requiere que cada articulación tenga algún tipo de dispositivo detector de posición.

El método más común es ubicar un sensor de posición directamente en el eje del actuador. Si el tren de transmisión es rígido y no tiene holgura, los ángulos de articulación reales pueden calcularse a partir de las posiciones del eje del actuador. Dichos pares de sensor y actuador **coubicados** son los más fáciles de controlar.

El dispositivo de retroalimentación de posición más popular es el **codificador óptico giratorio**. A medida que gira el eje del codificador, un disco que contiene un patrón de líneas interrumpe un haz de luz. Un fotodetector convierte estos pulsos de luz en una forma de onda binaria. Generalmente hay dos de tales canales, con trenes de pulso de onda desfasados 90 grados. El ángulo del eje se determina contando el número de pulsos, y la dirección de rotación se determina mediante la fase relativa de las dos ondas cuadradas. Además, los codificadores generalmente emiten un **pulso índice** en una ubicación que puede usarse para establecer una posición inicial y poder calcular una posición angular absoluta.

Los **resolventes** son dispositivos que producen dos señales análogas; una es el seno del ángulo del eje, la otra es el coseno. El ángulo del eje se calcula a partir de la magnitud relativa de las dos señales; la resolución es una función de la calidad del resolvente y la cantidad de ruido absorbido en los componentes electrónicos y el cableado. Los resolventes son a menudo más confiables que los codificadores ópticos, pero su resolución es menor. Generalmente los resolventes no pueden colocarse directamente en la articulación sin engranaje adicional para mejorar la resolución.

Los **potenciómetros** son la forma más simple de detección de posición. Al conectarse en configuración de puente producen un voltaje proporcional a la posición del eje; pero las dificultades con la resolución, linealidad y susceptibilidad al ruido limitan su uso.

Algunas veces, los **tacómetros** se utilizan para proporcionar una señal análoga proporcional a la velocidad del eje. En ausencia de dichos sensores de velocidad, la retroalimentación de velocidad se deriva tomando diferencias de la posición detectada en el tiempo. Esta **diferenciación numérica** puede introducir tanto ruido como un retraso de tiempo. A pesar de estos problemas potenciales, la mayoría de los manipuladores no tienen una detección directa de la velocidad.

8.9 DETECCIÓN DE FUERZA

Existe una gran variedad de dispositivos para medir las fuerzas de contacto entre el efector final de un manipulador y el entorno con el que interactúa. La mayoría de dichos sensores utiliza elementos de detección llamados **indicadores de tensión**, ya sea la variedad de semiconductor o la de laminilla de metal. Estos indicadores de tensión están enlazados a una estructura de metal y producen una salida proporcional a la tensión en el metal. En este tipo de diseño de detector de fuerza, los puntos con los que debe tratar el diseñador son:

1. ¿Cuántos sensores se necesitan para resolver la información deseada?
2. ¿Cómo están montados los sensores de forma relativa unos con otros en la estructura?
3. ¿Qué estructura permite una buena sensibilidad, manteniendo al mismo tiempo la rigidez?
4. ¿Cómo puede integrarse la protección contra sobrecarga mecánica en el dispositivo?

Hay tres lugares en los que generalmente se colocan esos sensores en un manipulador:

1. En los actuadores de las articulaciones. Estos sensores miden la salida del momento de torsión o fuerza del actuador/reducción. Son útiles para algunos esquemas de control, pero generalmente no proporcionan una buena detección del contacto entre el efector final y el entorno.
2. Entre el efector final y la última articulación del manipulador. Estos sensores, conocidos comúnmente como **sensores de muñeca**, son estructuras mecánicas equipadas con indicadores de tensión, los cuales pueden medir las fuerzas y momentos de torsión que actúan sobre el efector final. Generalmente estos sensores son capaces de medir de tres a seis componentes del vector de fuerza/momento de torsión que actúa sobre el efector final.
3. En la "punta de los dedos" del efector final. Generalmente estos **dedos detectores de fuerza** tienen indicadores de tensión integrados para medir de uno a cuatro componentes de fuerzas que actúen en la punta de cada dedo.

El dibujo de la figura 8.19 es un ejemplo de la estructura interna de un popular estilo de sensor de fuerza de muñeca diseñado por Scheinman [22]. Hay ocho pares de calibradores de deformación del tipo semiconductor enlazados a la estructura de barra cruzada del dispositivo. Cada par está cableado en un arreglo de divisor de voltaje. Cada vez que se consulta la muñeca, se digitalizan y leen en la computadora ocho voltajes análogos. Se han diseñado esquemas de calibración para llegar a una **matriz de calibración** constante de 6×8 que asigna estas ocho mediciones de tensión al vector de fuerza-momento de torsión $\mathcal{F}$, el cual actúa en el efector final. Como vimos en el ejemplo 5.8, el vector de fuerza-momento de torsión detectado puede transformarse en una trama de referencia que sea de interés.

Cuestiones de diseño del sensor de fuerza

El uso de calibradores de deformación para medir la fuerza se basa en la medición de la deflexión de un flexor tensionado. Por lo tanto, una de las principales restricciones de diseño está en elegir entre la rigidez y la sensibilidad del sensor. Un sensor más rígido es inherentemente menos sensible.

La rigidez del sensor también afecta la construcción de la **protección contra sobrecargas**. Los indicadores de tensión pueden dañarse mediante una sobrecarga por

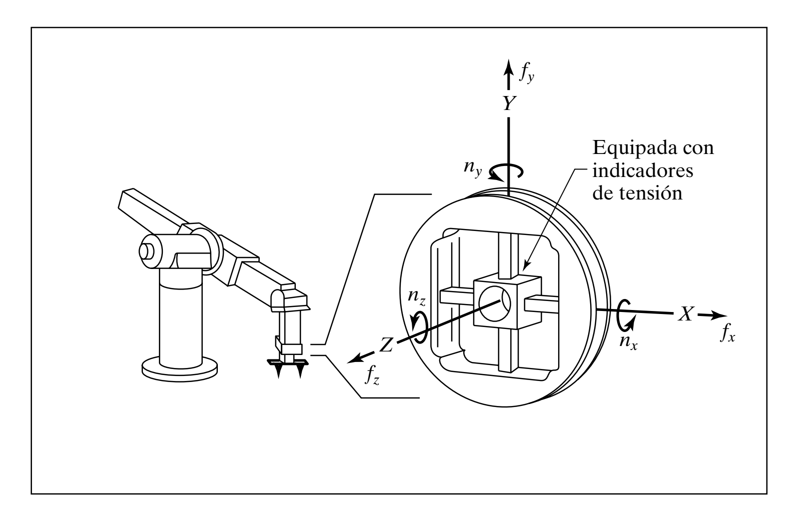


FIGURA 8.19: La estructura interna de una muñeca detectora de fuerza típica.

impacto, y por lo tanto deben protegerse contra dichas sobrecargas. El daño al transductor puede prevenirse mediante **topes delimitadores**, los cuales evitan que las flexiones se deflexionen más allá de cierto punto. Desafortunadamente, un sensor muy rígido podría deflexionarse solamente unas cuantas diezmilésimas de pulgada. Es muy difícil fabricar topes delimitadores con esos espacios libres tan pequeños. En consecuencia, para muchos transductores *debe* incorporarse cierto tipo de flexibilidad para poder crear posibles topes delimitadores efectivos.

La eliminación de la **histéresis** es una de las restricciones más molestas en el diseño de sensores. La mayoría de los metales utilizados como flexores tienen muy poca histéresis si no se deforman en exceso. No obstante, las articulaciones sujetas con pernos, a presión o soldadas cerca del flexor agregan histéresis. Lo ideal es que el flexor y el material cercano a él estén creados a partir de una sola pieza de metal.

Es importante utilizar medidas diferenciales para incrementar la linealidad y el rechazo de disturbios de los sensores de momento de torsión. Las distintas configuraciones físicas de los transductores pueden eliminar influencias debidas a los efectos de la temperatura y las fuerzas fuera del eje.

Los indicadores de laminilla de metal son relativamente resistentes, pero producen un pequeño cambio de resistencia a tensión completa. La eliminación de ruido en el cableado del indicador de tensión y en los componentes electrónicos de amplificación es de crucial importancia para obtener un buen rango dinámico.

Los indicadores de tensión tipo semiconductor son mucho más susceptibles al daño por sobrecarga, aunque tienen la ventaja de producir un cambio de resistencia de aproximadamente setenta veces el de los indicadores de laminilla de metal para una tensión dada. Esto hace que la tarea de procesamiento de señales sea mucho más simple para un rango dinámico dado.

BIBLIOGRAFÍA

[1] W. Rowe, Editor, *Robotics Technical Directory 1986*, Instrument Society of America, Research Triangle Park, NC, 1986.

[2] R. Vijaykumar y K. Waldron, “Geometric Optimization of Manipulator Structures for Working Volume and Dexterity”, *International Journal of Robotics Research*, Vol. 5, Núm. 2, 1986.

[3] K. Waldron, "Design of Arms", en *The International Encyclopedia of Robotics*, R. Dorf y S. Nof, Editores, John Wiley and Sons, Nueva York, 1988.

[4] T. Yoshikawa, "Manipulability of Robotic Mechanisms", *The International Journal of Robotics Research*, Vol. 4, Núm. 2, MIT Press, Cambridge, MA, 1985.

[5] H. Asada, "Dynamic Analysis and Design of Robot Manipulators Using Inertia Ellipsoids", *Proceedings of the IEEE International Conference on Robotics*, Atlanta, marzo de 1984.

[6] J. K. Salisbury y J. Craig, "Articulated Hands: Force Control and Kinematic Issues", *The International Journal of Robotics Research*, Vol. 1, Núm. 1, 1982.

[7] O. Khatib y J. Burdick, "Optimization of Dynamics in Manipulator Design: The Operational Space Formulation", *International Journal of Robotics and Automation*, Vol. 2, Núm. 2, IASTED, 1987.

[8] T. Yoshikawa, "Dynamic Manipulability of Robot Manipulators", *Proceedings of the IEEE International Conference on Robotics and Automation*, St. Louis, marzo de 1985.

[9] J. Trevelyan, P. Kovesi y M. Ong, "Motion Control for a Sheep Shearing Robot", *The 1st International Symoposium of Robotics Research*, MIT Press, Cambridge, MA, 1984.

[10] P. Marchal, J. Cornu y J. Detriche, "Self Adaptive Arc Welding Operation by Means of an Automatic Joint Following System", *Proceedings of the 4th Symposium on Theory and Practice of Robots and Manipulators*, Zaburow, Polonia, septiembre de 1981.

[11] J.M. Hollerbach, "Optimum Kinematic Design for a Seven Degree of Freedom Manipulator", *Proceedings of the 2nd International Symposium of Robotics Research*, Kyoto, Japón, agosto de 1984.

[12] K, Waldron y J. Reidy, "A Study of Kinematically Redundant Manipulator Structure", *Proceedings of the IEEE Robotics and Automation Conference*, San Francisco, abril de 1986.

[13] V. Milenkovic, "New Nonsingular Robot Wrist Design", *Proceedings of the Robots 11 / 17th ISIR Conference,* SME, 1987.

[14] E. Rivin, *Mechanical Design of Robots*, McGraw-Hill, Nueva York, 1988.

[15] T. Yoshikawa, "Manipulability of Robotic Mechanisms", en *Proceedings of the 2nd International Symposium on Robotics Research*, Kyoto, Japón, 1984.

[16] M. Leu, V. Dukowski y K. Wang, "An Analytical and Experimental Study of the Stiffness of Robot Manipulators with Parallel Mechanisms", en *Robotics and Manufacturing Automation*, M. Donath y M. Leu, Editores, ASME, Nueva York, 1985.

[17] K. Hunt, *Kinematic Geometry of Mechanisms*, Cambridge University Press, Cambridge, MA, 1978.

[18] H. Asada y K. Youcef-Youmi, *Design of Direct Drive Manipulators*, MIT Press, Cambridge, MA, 1987.

[19] J. Shigley, *Mechanical Engineering Design*, 3a. Edición, McGraw-Hill, Nueva York, 1977.

[20] D. Welbourne, "Fundamental Knowledge of Gear Noise—A Survey", *Proceedings of the Conference on Noise and Vibrations of Engines and Transmissions,* Instituto de Ingenieros Mecánicos, Cranfield, Reino Unido, 1979.

[21] O. Zienkiewuicz, *The Finite Element Method*, 3a. Edición, McGraw-Hill, Nueva York, 1977.

[22] V. Scheinman, "Design of a Computer Controlled Manipulator", tesis de maestría, Departamento de Ingeniería Mecánica, Universidad de Stanford, 1969.

[23] K. Lau, N. Dagalakis y D. Meyers, "Testing", en *The International Encyclopedia of Robotics*, R. Dorf y S. Nof, Editores, John Wiley and Sons, Nueva York, 1988.

[24] M. Roshiem, "Wrists", en *The International Encyclopedia of Robotics*, R. Dorf y S. Nof, Editores, John Wiley and Sons, Nueva York, 1988.

[25] A. Bowling y O. Khatib, "Robot Acceleration Capability: The Actuation Efficiency Measure", *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, abril de 2000.

EJERCICIOS

8.1 [15] Se va a utilizar un robot para posicionar un dispositivo de corte con láser. El láser produce un rayo de precisión no divergente. Para tareas de corte general, ¿cuántos grados de libertad necesita el robot de posicionamiento? Justifique su respuesta.

8.2 [15] Bosqueje una posible configuración de las articulaciones para el robot de posicionamiento de láser del ejercicio 8.1, suponiendo que se utilizará principalmente para cortar en ángulos particulares a través de placas de 8×8 pies, con 1 pulgada de grosor.

8.3 [17] Para un robot esférico como el de la figura 8.6, si las articulaciones 1 y 2 no tienen límites y la articulación 3 tiene un límite inferior l y un límite superior u, encuentre el índice de longitud estructural Q_L para el punto de la muñeca de este robot.

8.4 [25] Un eje de acero con una longitud de 30 cm y un diámetro de 0.2 cm controla el engranaje de entrada de una reducción con $\eta = 8$. El engranaje de salida controla un eje de acero que tiene una longitud de 30 cm y un diámetro de 0.3 cm. Si los engranajes no introducen deformación por sí solos, ¿cuál es la rigidez total del sistema de transmisión?

8.5 [20] En la figura 8.20 un vínculo se controla a través de un eje que va después de una reducción de engranajes. Modele el vínculo como rígido, con una masa de 10 Kg ubicada en un punto a 30 cm del eje del árbol. Suponga que los engranajes son rígidos y que la reducción η es grande. El eje es de acero y debe tener 30 cm de longitud. Si las especificaciones de diseño indican que el centro de la masa del vínculo debe soportar aceleraciones de 2.0 g, ¿cuál debería ser el diámetro del eje para limitar las deflexiones dinámicas a 0.1 radianes en el ángulo de la articulación?

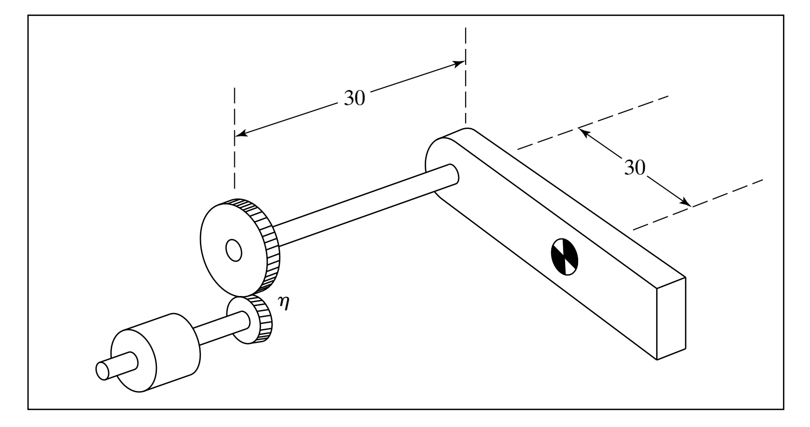


FIGURA 8.20: Un vínculo accionado a través de un eje que va después de una reducción de engranajes.

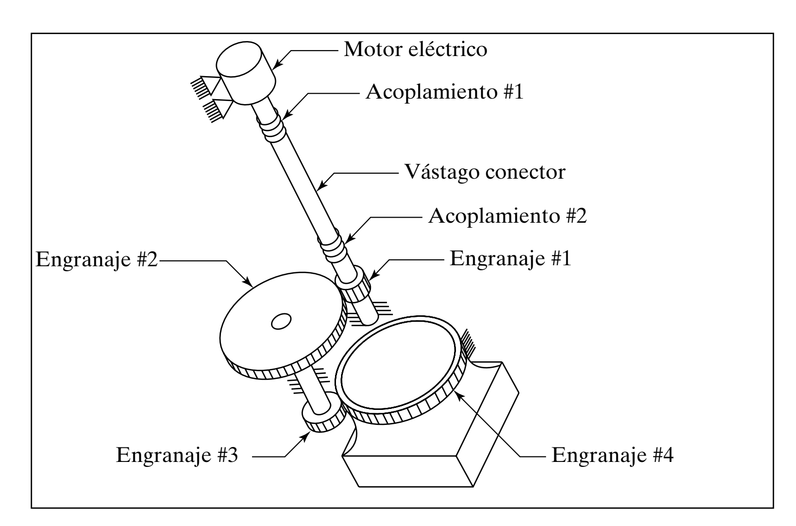


FIGURA 8.21: Versión simplificada del tren de transmisión de la articulación 4 del manipulador PUMA 560 (de [23]). De *International Encyclopedia of Robotics*, de R. Dorf y J. S. Nof, editores. De "Testing" de K. Law, N. Dagalakis y D. Myers.

8.6 [15] Si el engranaje de salida exhibe una rigidez de 1000 Nt-m/radián con el engranaje de entrada bloqueado y el árbol tiene una rigidez de 300 Nt-m/radián, ¿cuál es la rigidez combinada del sistema de transmisión en la figura 8.20?

8.7 [43] El criterio de Pieper para los manipuladores con vínculos seriales establece que el manipulador podrá resolverse si tres ejes consecutivos se intersecan en un solo punto o son paralelos. Esto se basa en la idea de que la cinemática inversa puede desacoplarse analizando la posición del punto de la muñeca, independientemente de la orientación de la trama de la muñeca. Proponga un resultado similar para el mecanismo Stewart de la figura 8.14, para que la solución cinemática directa pueda desacoplarse de manera similar.

8.8 [20] En el mecanismo Stewart de la figura 8.14, si las articulaciones universales 2-DOF en la base se sustituyeran por articulaciones cotiloideas 3-DOF, ¿cuál sería el número total de grados de libertad del mecanismo? Utilice la fórmula de Grübler.

8.9 [22] La figura 8.21 muestra un diagrama esquemático simplificado del sistema de transmisión de la articulación 4 del PUMA 560 [23]. La rigidez torsional de los acoplamientos es de 100 Nt-m/radián en cada uno, la del eje es de 400 Nt-m/radián, y cada uno de los pares de reducción se ha medido para tener una rigidez de salida de 2000 Nt-m/radián con sus engranajes de entrada fijos. Ambas reducciones tienen $\eta = 6$.[2] Suponiendo que la estructura y el cojinete son perfectamente rígidos, ¿cuál es la rigidez en la articulación (en particular, cuando el eje del motor está bloqueado)?

8.10 [25] ¿Cuál es el error si se aproxima la respuesta al ejercicio 8.9 considerando sólo la rigidez del engranaje final de reducción de velocidad?

8.11 [20] La figura 4.14 muestra una muñeca con eje ortogonal y otra con eje no ortogonal. La muñeca con eje ortogonal tiene torsiones de vínculo de una magnitud de 90°; la muñeca no ortogonal tiene torsiones de vínculo de ϕ (en magnitud) y 180°. Describa el conjunto de orientaciones que son *inalcanzables* con el mecanismo no ortogonal. Suponga que todos los ejes pueden girar 360° y que, de ser necesario, los vínculos pueden pasar unos a través de otros (es decir, el espacio de trabajo no está limitado por colisiones automáticas).

[2] Ninguno de los valores numéricos en este ejercicio es real.

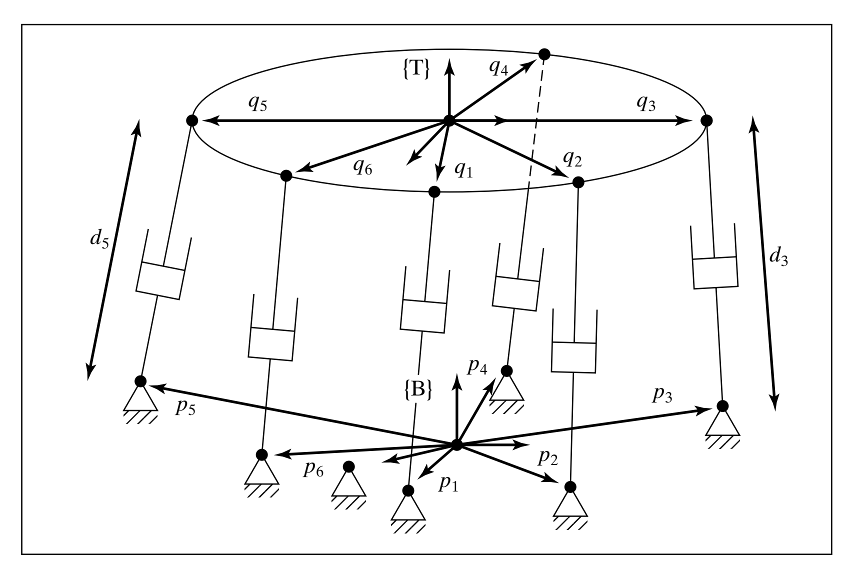


FIGURA 8.22: Mecanismo Stewart del ejercicio 8.12.

8.12 [18] Escriba una solución de cinemática inversa general para el mecanismo Stewart que se muestra en la figura 8.22. Dada la ubicación de $\{T\}$ relativa a la trama base $\{B\}$, resuelva para las variables de posición de articulación d_1 a d_6. Los ${}^{B}p_i$ son vectores de 3×1 que ubican las conexiones base de los actuadores lineales relativos a la trama $\{B\}$. Los ${}^{T}q_i$ son vectores de 3×1 que ubican las conexiones superiores de los actuadores lineales respecto a la trama $\{T\}$.

8.13 [20] El determinante jacobiano del manipulador planar de dos vínculos del ejemplo 5.3 se da mediante

$$\det(J(\Theta)) = l_1 l_2 s_2. \tag{8.31}$$

Si la suma de las dos longitudes de vínculo $l_1 + l_2$ se restringe para ser igual a una constante, ¿cuáles deberían ser las longitudes relativas para poder maximizar la medida de manipulabilidad del manipulador, según se definió en la ecuación (8.6)?

8.14 [28] Para un robot SCARA, dado que la suma de las longitudes de los vínculos 1 y 2 debe ser constante, ¿cuál es la elección óptima de longitud relativa en términos del índice de manipulabilidad presentado en la ecuación (8.6)? Podría ser útil resolver primero el ejercicio 8.13.

8.15 [35] Demuestre que la medida de manipulabilidad definida en la ecuación (8.6) también es igual al producto de los valores propios de $J(\Theta)$.

8.16 [15] ¿Cuál es la rigidez torsional de un vástago de aluminio de 40 cm con un radio de 0.1 cm?

8.17 [5] ¿Cuál es la reducción efectiva de "engranaje" η de un sistema de correa que tiene una polea de entrada con un radio de 2.0 cm y una polea de salida con un radio de 12.0 cm?

8.18 [10] ¿Cuántos grados de libertad se requieren en un manipulador que se utiliza para colocar piezas en forma de cilindro en un plano uniforme? Las piezas cilíndricas son perfectamente simétricas sobre sus ejes principales.

8.19 [25] La figura 8.23 muestra una mano con tres dedos sujetando un objeto. Cada dedo tiene tres articulaciones con un solo grado de libertad. Los puntos de contacto entre las puntas de los dedos y el objeto se modelan como "contacto de punto"; esto es, la posición es fija pero la orientación relativa es libre en los tres grados de libertad. Por

lo tanto, para fines de análisis estos contactos de puntos pueden sustituirse por articulaciones cotiloideas 3-DOF. Aplique la fórmula de Grübler para calcular cuántos grados de libertad posee el sistema en general.

8.20 [23] La figura 8.24 muestra un objeto conectado a tierra por medio de tres vástagos. Cada vástago está conectado al objeto con una articulación universal 2-DOF y a la tierra con una articulación cotiloidea 3-DOF. ¿Cuántos grados de libertad posee el sistema?

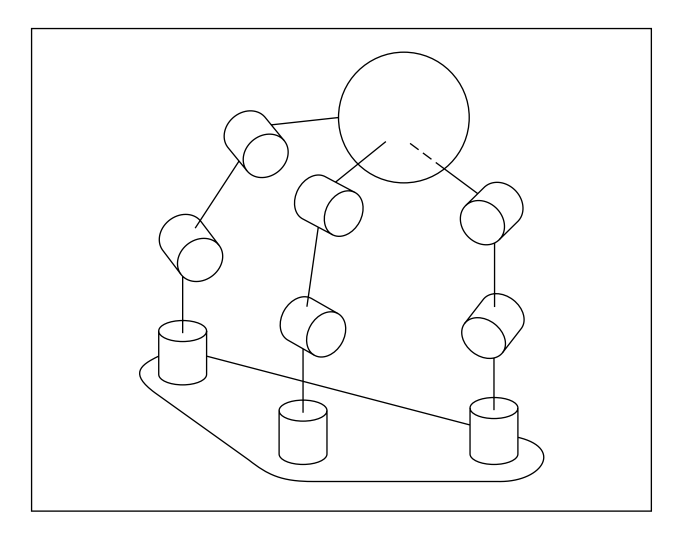


FIGURA 8.23: Una mano con tres dedos, en la que cada dedo tiene tres grados de libertad, atrapa un objeto con "contacto de punto".

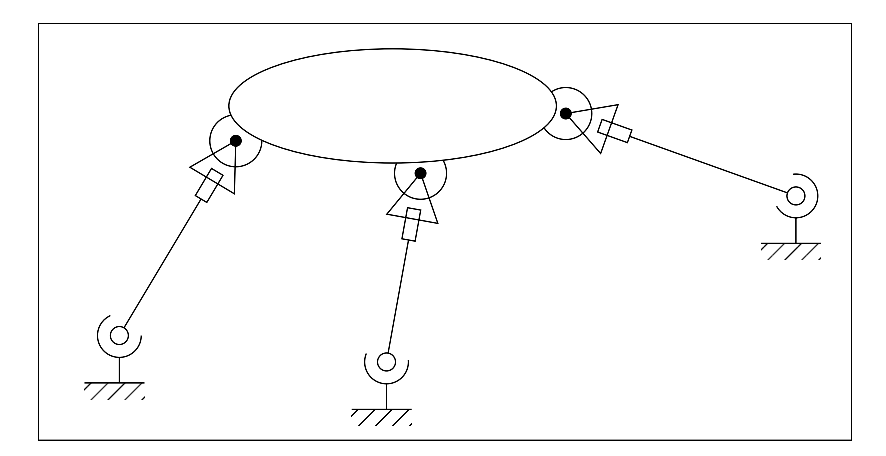


FIGURA 8.24: Mecanismo de lazo cerrado del ejercicio 8.20.

8.21 [18] Verifique que si dos sistemas de transmisión están conectados en forma serial, la rigidez total del sistema se debe obtener mediante la ecuación (8.14). Es tal vez más

simple pensar en la conexión serial de dos resortes lineales que tienen coeficientes de rigidez k_1 y k_2 y en las ecuaciones resultantes:

$$\begin{aligned} f &= k_1 \delta x_1, \\ f &= k_2 \delta x_2, \\ f &= k_{sum}(\delta x_1 + \delta x_2). \end{aligned} \tag{8.32}$$

8.21 [20] Derive una fórmula para la rigidez de un sistema de control de correa en términos de los radios de la polea (r_1 y r_2) y de la distancia de centro a centro entre las poleas, d_c. Empiece desde la ecuación (8.22).

EJERCICIO DE PROGRAMACIÓN (PARTE 8)

1. Escriba un programa para calcular la determinante de una matriz de 3 × 3.
2. Escriba un programa para mover el robot simulado de tres vínculos en 20 pasos en línea recta y una orientación constante, de

$${}^0_3T = \begin{bmatrix} 0.25 \\ 0.0 \\ 0.0 \end{bmatrix}$$

a

$${}^0_3T = \begin{bmatrix} 0.95 \\ 0.0 \\ 0.0 \end{bmatrix}$$

en incrementos de 0.05 metros. En cada ubicación, calcule la medida de manipulabilidad para el robot en esa configuración (es decir, el determinante del jacobiano). Liste o, mejor aún, haga un trazo de los valores como una función de la posición a lo largo del eje $\hat{X}_0$. Genere los datos anteriores para dos casos:

(a) $l_1 = l_2 = 0.5$ metros.
(b) $l_1 = 0.625$ metros, $l_2 = 0.375$ metros.

¿Cuál diseño de manipulador piensa usted que es mejor? Explique su respuesta.

EJERCICIO MATLAB 8

En la sección 8.5 presentamos el concepto de robots cinemáticamente redundantes. En este ejercicio trataremos la simulación de control con velocidad resuelta para un robot cinemáticamente redundante. Nos enfocaremos en el robot planar 4-DOF 4R con un grado de redundancia cinemática (cuatro articulaciones para proporcionar tres movimientos cartesianos: dos traslaciones y una rotación). Este robot se obtiene agregando una cuarta articulación R y un cuarto vínculo móvil L_4 al robot planar 3-DOF 3R (de las figuras 3.6 y 3.7; los parámetros de DH pueden extenderse si agregamos una fila a la figura 3.8).

Para el robot planar 4R derive expresiones analíticas para la matriz jacobiana de 3 × 4; después realice una simulación de control con velocidad resuelta en MATLAB (como en el ejercicio MATLAB 5). La forma de la ecuación de velocidad es, de nuevo, ${}^k\dot{X} = {}^kJ\dot{\Theta}$; no obstante, esta ecuación no puede invertirse mediante la inversa de matriz normal, ya que la matriz jacobiana no es cuadrada (tres ecuaciones, cuatro variables desconocidas, soluciones infinitas para $\dot{\Theta}$). Por lo tanto, vamos a utilizar la seudo inversa Moore-Penrose J^* de la matriz jacobiana: $J^* = J^T(JJ^T)^{-1}$. Para las velocidades de articulación relativas comandadas que resultan para el algoritmo de proporción resuelta, $\dot{\theta} = {}^kJ^*\,{}^k\dot{X}$, seleccione la solución de

norma mínima entre las posibilidades infinitas (es decir, esta $\dot{\Theta}$ específica es lo más pequeña posible para satisfacer las velocidades cartesianas ${}^{k}\dot{X}$ comandadas).

Esta solución representa solamente la solución particular; es decir, existe una solución homogénea para optimizar el rendimiento (como evitar las singularidades del manipulador o evitar los límites de las articulaciones), además de satisfacer el movimiento cartesiano comandado. (La optimización del rendimiento está más allá del alcance de este libro).

Dadas: $L_1 = 1.0\ m$, $L_2 = 1.0$ m, $L_3 = 0.2$ m, $L_4 = 0.2$ m.
Los ángulos iniciales son:

$$\Theta = \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{Bmatrix} = \begin{Bmatrix} -30^\circ \\ 70^\circ \\ 30^\circ \\ 40^\circ \end{Bmatrix}.$$

La velocidad cartesiana comandada (constante) es

$${}^{0}\dot{X} = {}^{0}\begin{Bmatrix} \dot{x} \\ \dot{y} \\ \omega_z \end{Bmatrix} = {}^{0}\begin{Bmatrix} -0.2 \\ -0.2 \\ 0.2 \end{Bmatrix} \text{ (m/s, rad/s).}$$

Simule el movimiento con velocidad resuelta para la solución particular solamente, durante 3 segundos, con un intervalo de control de 0.1 segundos. Además, en el mismo ciclo anime el robot en la pantalla durante cada intervalo, para que pueda observar el movimiento simulado y verificar que sea correcto.

a) Presente cuatro diagramas (cada conjunto en un gráfico separado, por favor):

1. Los cuatro ángulos de articulación (grados) $\Theta = \{\theta_1\ \theta_2\ \theta_3\ \theta_4\}^T$ contra el tiempo.
2. Las cuatro proporciones de articulación (rad/s) $\dot{\Theta} = \{\dot{\theta}_1\ \dot{\theta}_2\ \dot{\theta}_3\ \dot{\theta}_4\}^T$ contra el tiempo.
3. La norma euclidiana de proporción de articulación $\|\dot{\Theta}\|$ (magnitud vectorial) contra el tiempo.
4. Los tres componentes cartesianos de ${}^{0}_{H}T$, $X = \{x\ \ y\ \ \phi\}^T$ (radianes está bien para ϕ, de manera que se ajuste) contra el tiempo.

Etiquete cuidadosamente (a mano está bien) cada componente en cada diagrama. Además, etiquete los nombres de los ejes y las unidades.

b) Verifique los resultados de su matriz jacobiana para los conjuntos de ángulos inicial y final mediante la Herramienta de Robótica MATLAB de Corke. Pruebe la función *jacob0()*. **Precaución:** Las funciones jacobianas del cuadro de herramientas son para el movimiento de {4}respecto a {0}, no para {*H*} respecto a {0} como en la asignación del problema. La función anterior proporciona el resultado jacobiano en coordenadas de {0}; *jacobn()* daría los resultados en coordenadas de la trama {4}

C A P Í T U L O 9

Control lineal de los manipuladores

9.1 INTRODUCCIÓN

Armados con el material anterior, ahora tenemos los medios para calcular historiales de tiempo de la posición de las articulaciones que correspondan a los movimientos deseados del efector final a través del espacio. En este capítulo empezaremos por hablar sobre cómo hacer que el manipulador verdaderamente realice estos movimientos deseados.

Los métodos de control que describiremos caen en la clase llamada sistemas de **control lineal**. En sentido estricto, el uso de técnicas de control lineal es válido solamente cuando el sistema que se está estudiando puede modelarse matemáticamente mediante ecuaciones diferenciales *lineales*. Para el caso del control de manipuladores, dichos métodos lineales deben verse como métodos aproximados ya que, como vimos en el capítulo 6, la dinámica de un manipulador se representa más adecuadamente mediante una ecuación diferencial *no lineal*. Sin embargo, veremos que a menudo es razonable hacer dichas aproximaciones, y también es el caso de que estos métodos lineales son los que se utilizan con más frecuencia en la práctica industrial actual.

Por último, la consideración del método lineal servirá como base para el tratamiento más complejo de los sistemas de control no lineales que veremos en el capítulo 10. Aunque nos acercamos al control lineal como un método aproximado para el control de manipuladores, la justificación de usar controladores lineales no solamente es empírica. En el capítulo 10 demostraremos que cierto controlador lineal produce un sistema de control razonable, incluso sin requerir una aproximación lineal de la dinámica del manipulador. Tal vez los lectores familiarizados con sistemas de control lineal quieran saltarse las primeras cuatro secciones del presente capítulo.

9.2 RETROALIMENTACIÓN Y CONTROL DE LAZO CERRADO

Vamos a modelar un manipulador como un mecanismo equipado con sensores en cada articulación para medir el ángulo de articulación, y que tiene un actuador en cada articulación para aplicar un momento de torsión sobre el vínculo adyacente (de mayor orden).[1] Aunque algunas veces se utilizan otros arreglos físicos de sensores, la gran mayoría de los robots tienen un sensor de posición en cada articulación. Algunas veces también se utilizan sensores de velocidad (tacómetros) en las articulaciones. Hay varios esquemas de actuación y transmisión que prevalecen en los robots industriales, pero muchos de ellos pueden modelarse suponiendo que hay un solo actuador en cada articulación.

Deseamos hacer que las articulaciones del manipulador sigan trayectorias de posición prescritas, pero los actuadores son comandados en términos de momento de torsión, por lo que debemos utilizar algún tipo de **sistema de control** para calcular los comandos apropiados del actuador que realicen este movimiento deseado. Casi siempre estos momentos de torsión están determinados por el uso de **retroalimentación** de los sensores de las articulaciones para calcular el momento de torsión requerido.

La figura 9.1 muestra la relación entre el generador de trayectorias y el robot físico. El robot acepta un vector de momentos de torsión τ del sistema de control. Los sensores del manipulador permiten al controlador leer los vectores de las posiciones de las articulaciones, Θ, y las velocidades de las articulaciones, $\dot{\Theta}$. Todas las líneas de señal en la figura 9.1 llevan vectores de $N \times 1$ (en donde N es el número de articulaciones en el manipulador).

Consideremos qué algoritmo podría implementarse en el bloque etiquetado como "sistema de control" en la figura 9.1. Una posibilidad sería usar la ecuación dinámica del robot (según lo estudiado en el capítulo 6) para calcular los momentos de torsión requeridos para una trayectoria específica. El generador de trayectorias nos proporciona Θ_d, $\dot{\Theta}_d$ y $\ddot{\Theta}_d$, por lo que podríamos usar la ecuación (6.59) para calcular

$$\tau = M(\Theta_d)\ddot{\Theta}_d + V(\Theta_d, \dot{\Theta}_d) + G(\Theta_d). \tag{9.1}$$

Esto nos da los momentos de torsión que nuestro modelo indica que serían requeridos para realizar la trayectoria deseada. Si nuestro modelo dinámico fuera completo y preciso y no hubiera "ruido" u otras perturbaciones, el uso continuo de la ecuación (9.1) sobre la trayectoria deseada realizaría la trayectoria deseada. Desafortunadamente, la imperfección en el modelo dinámico y la inevitable presencia de disturbios hacen que un esquema así sea poco práctico para utilizarse en aplicaciones reales. A una técnica de control así se le denomina esquema de **lazo abierto**, ya que no se utiliza la retroalimentación de los sensores de las articulaciones [es decir, la ecuación (9.1) es una función

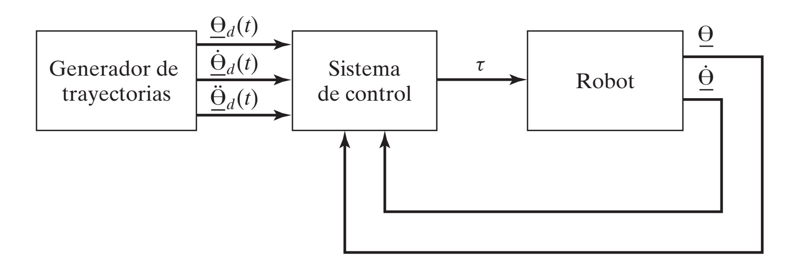


FIGURA 9.1: Diagrama de bloques de alto nivel de un sistema de control de robot.

[1] Recuerde que todos los comentarios que se hagan respecto a las articulaciones giratorias son análogos para las articulaciones lineales, y viceversa.

sólo de la trayectoria deseada, representada por Θ_d, y de sus derivadas, y *no es* una función de Θ, la trayectoria actual].

Generalmente, la única forma de construir un sistema de control de alto rendimiento es utilizando la retroalimentación de los sensores de las articulaciones, como se indica en la figura 9.1. Esta retroalimentación se utiliza comúnmente para calcular cualquier **error de servo**, encontrando la diferencia entre la posición deseada y la posición actual, y la diferencia entre la velocidad deseada y la velocidad actual:

$$E = \Theta_d - \Theta,$$
$$\dot{E} = \dot{\Theta}_d - \dot{\Theta}. \tag{9.2}$$

Así, el sistema de control puede calcular cuánto momento de torsión se requiere de los actuadores como alguna función del error de servo. Obviamente, la idea básica es calcular momentos de torsión de los actuadores que tiendan a reducir los errores de servo. A un sistema de control que utiliza retroalimentación se le llama sistema de **lazo cerrado**. En la figura 9.1 se aprecia el "lazo" cerrado alrededor del manipulador por un sistema de control de ese tipo.

El problema central al diseñar un sistema de control es asegurar que el sistema de lazo cerrado resultante cumpla con ciertas especificaciones de rendimiento. El criterio más básico de este tipo es que el sistema permanezca **estable**. Para nuestros fines, definiremos a un sistema como estable si los errores permanecen "pequeños" al ejecutar varias trayectorias deseadas, incluso en la presencia de ciertas perturbaciones "moderadas". Hay que tener en cuenta que un sistema de control diseñado inapropiadamente puede algunas veces producir una operación **inestable**, en donde los errores de servo se agrandan en vez de reducirse. Por ende, la primera tarea de un ingeniero de control es demostrar que su diseño produce un sistema estable; la segunda tarea es demostrar que el rendimiento de lazo cerrado del sistema es satisfactorio. En la práctica, dichas "pruebas" varían desde las pruebas matemáticas basadas en ciertas suposiciones y modelos hasta los resultados más empíricos, tales como los que se obtienen a través de la simulación o la experimentación.

La figura 9.1, en la que todas las líneas de señal representan vectores de $N \times 1$, resume el hecho de que el problema de control de los manipuladores es un problema de control **multientradas, multisalidas (MIMO)**. En este capítulo usaremos un método simple para construir un sistema de control, tratando cada articulación como un sistema que debe controlarse por separado. En consecuencia, para un manipulador con N articulaciones diseñaremos N sistemas independientes de control de **una entrada, una salida (SISO)**. Éste es el método de diseño adoptado actualmente por la mayoría de los proveedores de robots industriales. Este método de **control independiente de las articulaciones** es un método aproximado, ya que las ecuaciones de movimiento (desarrolladas en el capítulo 6) no son independientes, sino que tienen muchos acoplamientos. Más adelante presentaremos una justificación para el método lineal, cuando menos para el caso de los manipuladores con muchos engranajes.

9.3 SISTEMAS LINEALES DE SEGUNDO ORDEN

Antes de considerar el problema de control de los manipuladores, retrocedamos y empecemos por considerar un sistema mecánico simple. La figura 9.2 muestra un bloque de masa m unido a un resorte con una rigidez k y sujeto a una fricción de coeficiente b. La figura 9.2 también indica la posición cero y el sentido positivo de x, la posición del bloque. Suponiendo una fuerza de fricción proporcional a la velocidad del bloque, un diagrama de cuerpo libre de las fuerzas que actúan sobre el bloque nos conduce directamente a la ecuación de movimiento,

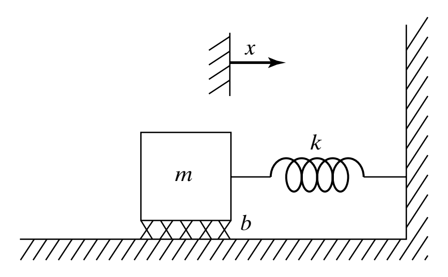


FIGURA 9.2: Sistema de resorte-masa con fricción.

$$m\ddot{x} + b\dot{x} + kx = 0. \tag{9.3}$$

Por consecuencia, la dinámica de lazo abierto de este sistema con un grado de libertad se describe mediante una ecuación diferencial lineal, de coeficientes constantes y segundo orden [1]. La solución a la ecuación diferencial (9.3) es una función del tiempo, $x(t)$, que especifica el movimiento del bloque. Esta solución dependerá de las **condiciones iniciales** del bloque, es decir, de su posición y su velocidad iniciales.

Utilizaremos este simple sistema mecánico como un ejemplo con el cual podremos repasar ciertos conceptos básicos de un sistema de control. Desafortunadamente, es imposible hacer justicia al campo de la teoría de control con sólo una breve introducción. Hablaremos sobre el problema de control, asumiendo que el estudiante está familiarizado con las ecuaciones diferenciales simples. Por ende, no utilizaremos muchas de las populares herramientas del área de ingeniería de control. Por ejemplo, las **transformadas de Laplace** y otras técnicas comunes no son un prerrequisito ni se presentan aquí. Una buena referencia para este campo es [4].

La intuición sugiere que el sistema de la figura 9.2 podría exhibir varios movimientos característicos distintos. Por ejemplo, en el caso de un resorte muy débil (es decir, con un valor de k pequeño) y una fricción muy pesada (es decir, un valor de b grande) cualquiera podría imaginarse que si el bloque fuera perturbado regresaría a su posición de descanso muy lentamente. Sin embargo, con un resorte muy rígido y una fricción muy baja, el bloque podría oscilar varias veces antes de quedar en reposo. Estas distintas posibilidades surgen debido a que el carácter de la solución a la ecuación (9.3) depende de los valores de los parámetros m, b y k.

Del estudio de las ecuaciones diferenciales [1] sabemos que la forma de la solución a una ecuación de la forma de la ecuación (9.3) depende de las raíces de su **ecuación característica**,

$$ms^2 + bs + k = 0. \tag{9.4}$$

Esta ecuación tiene las raíces

$$\begin{aligned} s_1 &= -\frac{b}{2m} + \frac{\sqrt{b^2 - 4mk}}{2m}, \\ s_2 &= -\frac{b}{2m} - \frac{\sqrt{b^2 - 4mk}}{2m}. \end{aligned} \tag{9.5}$$

La ubicación de s_1 y s_2 (algunas veces llamados los **polos** del sistema) en el plano real-imaginario dicta la naturaleza de los movimientos del sistema. Si s_1 y s_2 son reales, entonces el comportamiento del sistema es lento y no oscilatorio. Si s_1 y s_2 son complejos (es decir, si tienen un componente imaginario) entonces el comportamiento del sistema

es oscilatorio. Si incluimos el caso limitante especial entre estos dos comportamientos, tenemos tres clases de respuesta por estudiar:

1. **Raíces reales y desiguales.** Éste es el caso cuando $b^2 > 4\,mk$; esto es, la fricción domina y se produce un comportamiento lento. A esta respuesta se le llama **sobreamortiguada**.
2. **Raíces complejas.** Éste es el caso cuando $b^2 < 4\,mk$; esto es, la rigidez domina y se produce un comportamiento oscilatorio. A esta respuesta se le llama **subamortiguada**.
3. **Raíces reales e iguales.** Éste es el caso especial cuando $b^2 = 4\,mk$; esto es, la fricción y la rigidez están "balanceadas", lo que produce la respuesta no oscilatoria más rápida posible. A esta respuesta se le llama **críticamente amortiguada**.

El tercer caso (amortiguamiento crítico) es generalmente una situación deseable: el sistema anula las condiciones iniciales distintas de cero y regresa a su posición nominal lo más rápido posible, pero sin comportamiento oscilatorio.

Raíces reales y desiguales

Puede demostrarse fácilmente [por sustitución directa en la ecuación (9.3)] que la solución $x(t)$, que proporciona el movimiento del bloque en el caso de raíces reales y desiguales, tiene la forma

$$x(t) = c_1 e^{s_1 t} + c_2 e^{s_2 t}, \tag{9.6}$$

en donde s_1 y s_2 se dan mediante la ecuación (9.5). Los coeficientes c_1 y c_2 son constantes que pueden calcularse para cualquier conjunto dado de condiciones iniciales (es decir, la posición y la velocidad iniciales del bloque).

La figura 9.3 muestra un ejemplo de las ubicaciones de los polos y la correspondiente respuesta en el tiempo para una condición inicial distinta de cero. Cuando los polos de un sistema de segundo orden son reales y desiguales, el sistema exhibe un movimiento lento o sobreamortiguado.

En los casos en que uno de los polos tiene una magnitud mucho mayor que la otra, el polo de mayor magnitud puede ignorarse, ya que el término que corresponde a éste disminuirá a cero rápidamente, en comparación con el otro **polo dominante**. Esta misma noción de la dominación se extiende a los sistemas de mayor orden; por ejemplo, a

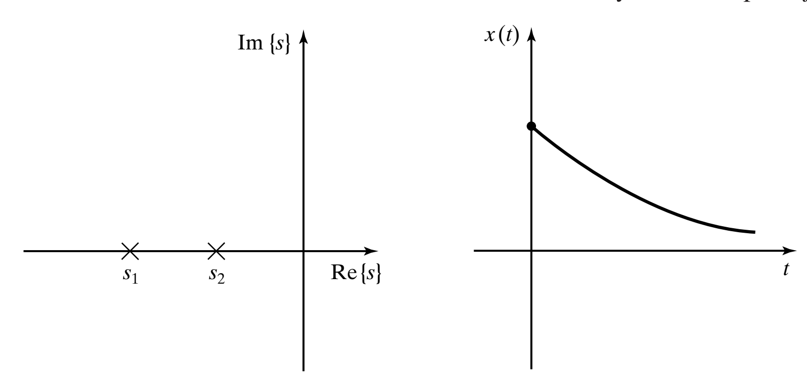


FIGURA 9.3: Ubicación de la raíz y la respuesta a las condiciones iniciales para un sistema sobreamortiguado.

menudo un sistema de tercer orden puede estudiarse como sistema de segundo orden si se consideran solamente los dos polos dominantes.

EJEMPLO 9.1

Determine el movimiento del sistema de la figura 9.2 si los valores de los parámetros son $m = 1$, $b = 5$ y $k = 6$, y si el bloque (inicialmente en reposo) se libera desde la posición $x = -1$.

La ecuación característica es

$$s^2 + 5s + 6 = 0, \tag{9.7}$$

la cual tiene las raíces $s_1 = -2$ y $s_2 = -3$. Por consecuencia, la respuesta tiene la forma

$$x(t) = c_1 e^{-2t} + c_2 e^{-3t}. \tag{9.8}$$

Ahora vamos a utilizar las condiciones iniciales dadas, $x(0) = -1$ y $x(0) = 0$, para calcular c_1 y c_2. Para satisfacer estas condiciones cuando $t = 0$, debemos tener

$$c_1 + c_2 = -1$$

y

$$-2c_1 - 3c_2 = 0, \tag{9.9}$$

las cuales se satisfacen cuando $c_1 = -3$ y $c_2 = 2$. Así, el movimiento del sistema para $t \geq 0$ se da mediante

$$x(t) = -3e^{-2t} + 2e^{-3t}. \tag{9.10}$$

Raíces complejas

Para el caso en el que la ecuación característica tiene raíces complejas de la forma

$$s_1 = \lambda + \mu i,$$
$$s_2 = \lambda - \mu i, \tag{9.11}$$

la solución sigue teniendo la forma

$$x(t) = c_1 e^{s_1 t} + c_2 e^{s_2 t}. \tag{9.12}$$

No obstante, la ecuación (9.12) es difícil de usar directamente, ya que involucra explícitamente a los números imaginarios. Puede demostrarse (vea el ejercicio 9.1) que la **fórmula de Euler**,

$$e^{ix} = \cos x + i \operatorname{sen} x, \tag{9.13}$$

permite que la solución (9.12) se manipule para expresarse en la forma

$$x(t) = c_1 e^{\lambda t} \cos(\mu t) + c_2 e^{\lambda t} \operatorname{sen}(\mu t). \tag{9.14}$$

Como antes, los coeficientes c_1 y c_2 son constantes que pueden calcularse para cualquier conjunto dado de condiciones iniciales (es decir, la posición y la velocidad iniciales del bloque). Si escribimos las constantes c_1 y c_2 en la forma

$$c_1 = r \cos \delta,$$
$$c_2 = r \operatorname{sen} \delta, \tag{9.15}$$

entonces la ecuación (9.14) puede escribirse en la forma

$$x(t) = re^{\lambda t}\cos(\mu t - \delta), \tag{9.16}$$

en donde

$$r = \sqrt{c_1^2 + c_2^2},$$
$$\delta = \text{Atan2}(c_2, c_1). \tag{9.17}$$

En esta forma es más sencillo ver que el movimiento resultante es una oscilación cuya amplitud está disminuyendo exponencialmente hacia cero.

Otra forma común de describir los sistemas oscilatorios de segundo orden es en términos de la **constante de amortiguamiento** y la **frecuencia natural**. Estos términos se definen por la parametrización de la ecuación característica, dada por

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0, \tag{9.18}$$

en donde ζ es la constante de amortiguamiento (un número sin dimensión, entre 0 y 1) y ω_n es la frecuencia natural.[2] Las relaciones entre las ubicaciones de los polos y estos parámetros son

$$\lambda = -\zeta\omega_n$$

y

$$\mu = \omega_n\sqrt{1-\zeta^2}. \tag{9.19}$$

En esta terminología, μ, la parte imaginaria de los polos, se conoce algunas veces como la **frecuencia natural amortiguada**. Para un sistema de resorte–masa amortiguado tal como el de la figura 9.2, la proporción de amortiguamiento y la frecuencia natural son, respectivamente,

$$\zeta = \frac{b}{2\sqrt{km}},$$
$$\omega_n = \sqrt{k/m}. \tag{9.20}$$

Cuando no hay amortiguamiento presente ($b = 0$ en nuestro ejemplo), la constante de amortiguamiento se vuelve cero; para un amortiguamiento crítico ($b^2 = 4km$), la constante es 1.

La figura 9.4 muestra un ejemplo de las ubicaciones de los polos y la correspondiente respuesta en el tiempo para una condición inicial distinta de cero. Cuando los polos de un sistema de segundo orden son complejos, el sistema exhibe un movimiento oscilatorio o subamortiguado.

EJEMPLO 9.2

Encuentre el movimiento del sistema en la figura 9.2 si los valores de los parámetros son $m = 1$, $b = 1$ y $k = 1$, y si el bloque (inicialmente en reposo) se libera desde la posición $x = -1$.

La ecuación característica es

$$s^2 + s + 1 = 0, \tag{9.21}$$

---

[2] Los términos "constante de amortiguamiento" y "frecuencia natural" también pueden aplicarse a sistemas con alto amortiguamiento, en cuyo caso $\zeta > 1.0$.

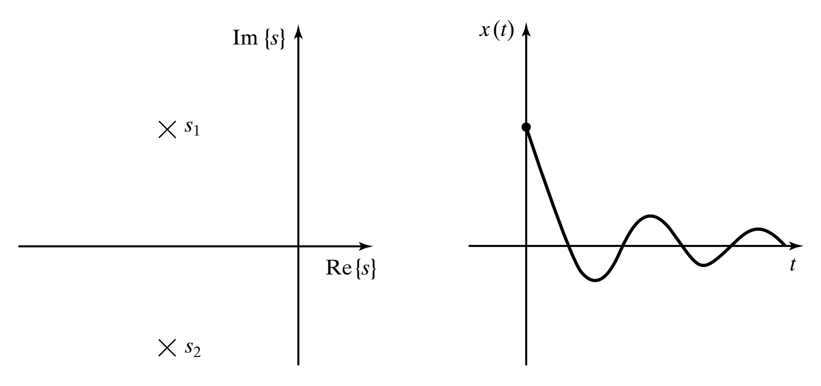


FIGURA 9.4: Ubicación de la raíz y respuesta a las condiciones iniciales para un sistema subamortiguado.

la cual tiene las raíces $s_i = -\frac{1}{2} \pm \frac{\sqrt{3}}{2} i$. Por lo tanto, la respuesta tiene la forma

$$x(t) = e^{-\frac{t}{2}} \left(c_1 \cos \frac{\sqrt{3}}{2} t + c_2 \operatorname{sen} \frac{\sqrt{3}}{2} t \right). \tag{9.22}$$

Ahora utilizamos las condiciones iniciales dadas, $x(0) = -1$ y $\dot{x}(0) = 0$, para calcular c_1 *y* c_2. Para satisfacer estas condiciones cuando $t = 0$, debemos tener

$$c_1 = -1$$

y

$$-\frac{1}{2} c_1 - \frac{\sqrt{3}}{2} c_2 = 0, \tag{9.23}$$

las cuales se satisfacen cuando $c_1 = -1$ y $c_2 = \frac{\sqrt{3}}{3}$. Por lo tanto, el movimiento del sistema para $t \geq 0$ se da mediante

$$x(t) = e^{-\frac{t}{2}} \left(-\cos \frac{\sqrt{3}}{2} t - \frac{\sqrt{3}}{3} \operatorname{sen} \frac{\sqrt{3}}{2} t \right). \tag{9.24}$$

Este resultado también puede ponerse en la forma de la ecuación (9.16):

$$x(t) = \frac{2\sqrt{3}}{3} e^{-\frac{t}{2}} \cos \left(\frac{\sqrt{3}}{2} t + 120° \right). \tag{9.25}$$

Raíces reales e iguales

Por sustitución, en la ecuación (9.3) puede demostrarse que, en el caso de raíces reales e iguales (es decir, **raíces repetidas**), la solución tiene la forma

$$x(t) = c_1 e^{s_1 t} + c_2 t e^{s_2 t}, \tag{9.26}$$

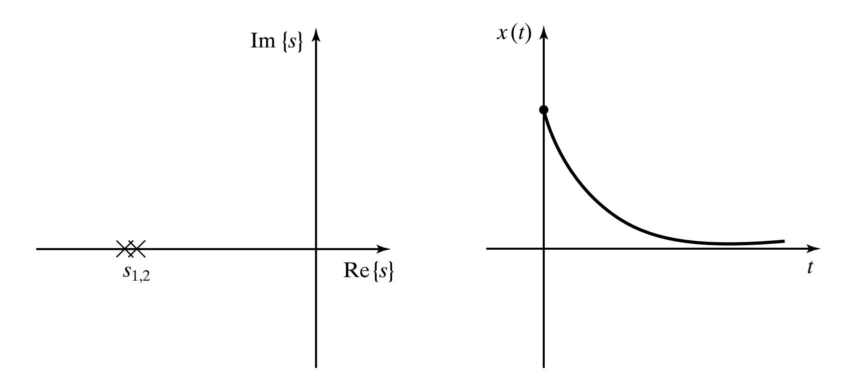


FIGURA 9.5: Ubicación de la raíz y respuesta a las condiciones iniciales para un sistema críticamente amortiguado.

en donde, en este caso, $s_1 = s_2 = -\frac{b}{2m}$, por lo que la ecuación (9.29) puede escribirse así:

$$x(t) = (c_1 + c_2 t)e^{-\frac{b}{2m}t}. \tag{9.27}$$

En caso de que no quede claro, una rápida aplicación de la **regla de l'Hopital** [2] muestra que, para cualquier valor de c_1, c_2 y a,

$$\lim_{t\to\infty} (c_1 + c_2 t)e^{-at} = 0. \tag{9.28}$$

La figura 9.5 muestra un ejemplo de las ubicaciones de los polos y la correspondiente respuesta en el tiempo para una condición inicial distinta de cero. Cuando los polos de un sistema de segundo orden son reales e iguales, el sistema exhibe un movimiento críticamente amortiguado, la respuesta no oscilatoria más rápida posible.

EJEMPLO 9.3

Encuentre el movimiento del sistema de la figura 9.2 si los valores de los parámetros son $m = 1$, $b = 4$ y $k = 4$, y si el bloque (inicialmente en reposo) se libera desde la posición $x = -1$.

La ecuación característica es

$$s^2 + 4s + 4 = 0, \tag{9.29}$$

la cual tiene las raíces $s_1 = s_2 = -2$. Por lo tanto, la respuesta tiene la forma

$$x(t) = (c_1 + c_2 t)e^{-2t}. \tag{9.30}$$

Ahora vamos a utilizar las condiciones iniciales dadas, $x(0) = -1$ y $\dot{x}(0) = 0$, para calcular c_1 y c_2. Para satisfacer estas condiciones cuando $t = 0$, debemos tener

$$c_1 = -1$$

y

$$-2c_1 + c_2 = 0, \tag{9.31}$$

las cuales se satisfacen cuando $c_1 = -1$ y $c_2 = -2$. Por lo tanto, el movimiento del sistema para $t \geq 0$ se da mediante

$$x(t) = (-1 - 2t)e^{-2t}. \tag{9.32}$$

En los ejemplos del 9.1 a 9.3, todos los sistemas eran estables. Para cualquier sistema físico pasivo como el de la figura 9.2, éste será el caso. Dichos sistemas mecánicos siempre tienen las propiedades

$$\begin{aligned} m &> 0, \\ b &> 0, \\ k &> 0. \end{aligned} \tag{9.33}$$

En la siguiente sección veremos que la acción de un sistema de control es, en efecto, cambiar el valor de uno o más de estos coeficientes; será necesario considerar si el sistema resultante es estable o no.

9.4 CONTROL DE SISTEMAS DE SEGUNDO ORDEN

Suponga que la respuesta natural de nuestro sistema mecánico de segundo orden no es lo que deseamos. Tal vez tenga bajo amortiguamiento y sea oscilatorio, y nos gustaría que fuera críticamente amortiguado; o tal vez no tenga un resorte ($k = 0$) por lo que el sistema nunca regresa a $x = 0$ si se le perturba. Mediante el uso de sensores, un actuador y un sistema de control podemos modificar el comportamiento del sistema según lo deseemos.

La figura 9.6 muestra un sistema de resorte–masa amortiguado al que se agregó un actuador, con el cual es posible aplicar una fuerza f al bloque. Un diagrama de cuerpo libre nos conduce a la ecuación de movimiento,

$$m\ddot{x} + b\dot{x} + kx = f. \tag{9.34}$$

Vamos a suponer también que tenemos sensores capaces de detectar la posición y la velocidad del bloque. Ahora propondremos una **ley de control**, la cual calcula la fuerza que el actuador debe aplicar como una función de la retroalimentación detectada:

$$f = -k_p x - k_v \dot{x}. \tag{9.35}$$

La figura 9.7 es un diagrama de bloques del sistema de lazo cerrado, en donde la porción a la izquierda de la línea punteada es el sistema de control (que generalmente se implementa en una computadora) y a la derecha de la línea punteada se encuentra el sistema físico. En la figura están implícitas las interfaces entre la computadora de control y los comandos del actuador de salida, y la información del sensor de entrada.

El sistema de control que hemos propuesto es un **sistema regulador de posición**; simplemente trata de mantener la posición del bloque en un lugar fijo, sin importar qué

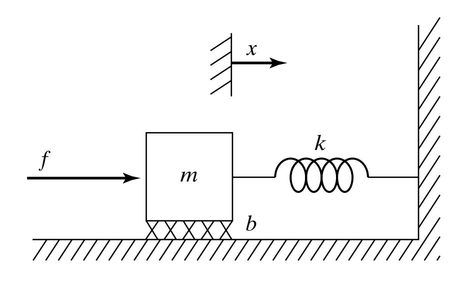


FIGURA 9.6: Un sistema de resorte–masa amortiguado, con un actuador.

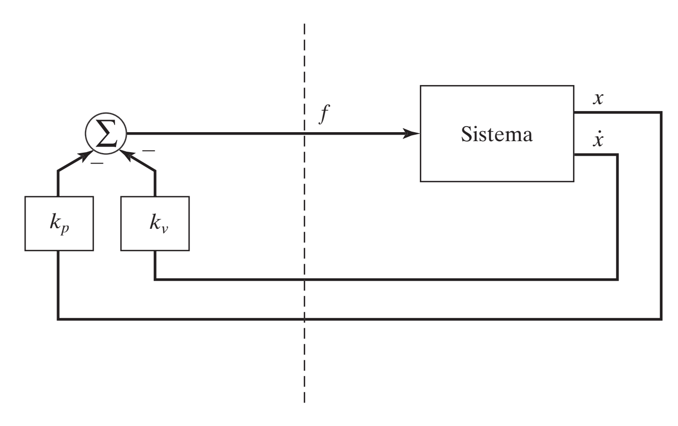


FIGURA 9.7: Un sistema de control de lazo cerrado. La computadora de control (a la izquierda de la línea punteada) lee la entrada del sensor y escribe los comandos de salida del actuador.

fuerzas perturbadoras se apliquen al bloque. En una sección posterior construiremos un sistema de control **de seguimiento de trayectorias**, el cual puede hacer que el bloque siga una trayectoria de posiciones deseadas.

Al igualar la dinámica de lazo abierto de la ecuación (9.34) con la ley de control de la ecuación (9.35), podemos derivar la dinámica de lazo cerrado así:

$$m\ddot{x} + b\dot{x} + kx = -k_p x - k_v \dot{x}, \tag{9.36}$$

o

$$m\ddot{x} + (b + k_v)\dot{x} + (k + k_p)x = 0, \tag{9.37}$$

o

$$m\ddot{x} + b'\dot{x} + k'x = 0, \tag{9.38}$$

en donde $b' = b + k_v$ y $k' = k + k_p$. De las ecuaciones (9.37) y (9.38) queda claro que, al establecer las **ganancias de control** k_v y k_p podemos hacer que el sistema de lazo cerrado parezca tener el comportamiento de cualquier sistema de segundo orden que deseemos. A menudo se seleccionan las ganancias para obtener un amortiguamiento crítico (es decir, $b' = 2\sqrt{mk'}$) y cierta **rigidez de lazo cerrado** proporcionada directamente por k'.

Observe que k_v y k_p podrían ser positivas o negativas, dependiendo de los parámetros del sistema original. No obstante, si b' o k' se vuelven negativas, el resultado sería un sistema de control inestable. Esta inestabilidad será obvia si se escribe la solución de la ecuación diferencial de segundo orden [en la forma de las ecuaciones (9.6), (9.14) o (9.26)]. También tiene sentido intuitivo que, si b' o k' son negativas, los errores de servo tiendan a magnificarse en vez de reducirse.

EJEMPLO 9.4

Si los parámetros del sistema de la figura 9.6 son $m = 1$, $b = 1$ y $k = 1$, encuentre las ganancias k_p y k_v para una ley de control de regulación de la posición que haga que el sistema esté críticamente amortiguado con una rigidez de lazo cerrado de 16.0.

Si deseamos que k' sea 16.0, entonces para un amortiguamiento crítico necesitamos que $b' = 2\sqrt{mk'} = 8.0$. Ahora, $k = 1$ y $b = 1$, por lo que necesitamos que

$$k_p = 15.0,$$
$$k_v = 7.0. \tag{9.39}$$

9.5 PARTICIONAMIENTO DE LEYES DE CONTROL

Para diseñar leyes de control para sistemas más complicados, vamos a considerar una estructura de controlador ligeramente distinta para el problema de ejemplo de la figura 9.6. En este método particionaremos el controlador en una **porción basada en el modelo** y en una **porción de servo**. Como resultado los parámetros del sistema (es decir, m, b y k en este caso) aparecen solamente en la porción basada en el modelo, y la porción de servo es independiente de estos parámetros. Hasta ahora esta distinción podría no parecer relevante, pero su importancia se hará obvia a medida que consideremos los sistemas no lineales en el capítulo 10. Adoptaremos este método de **particionamiento de las leyes de control** en todo el libro.

La ecuación de lazo abierto del movimiento para el sistema es

$$m\ddot{x} + b\dot{x} + kx = f. \tag{9.40}$$

Deseamos descomponer el controlador para este sistema en dos partes. En este caso, la porción de la ley de control basada en el modelo utilizará el supuesto conocimiento de m, b y k. Esta porción de la ley de control se establece de manera tal que *reduzca el sistema para que parezca una masa unitaria*. Esto nos quedará más claro cuando hagamos el ejemplo 9.5. La segunda parte de la ley de control hace uso de la retroalimentación para modificar el comportamiento del sistema. La porción de la ley de control basada en el modelo tiene el efecto de hacer que el sistema aparezca como una masa unitaria, por lo que el diseño de la porción de servo es muy simple: se seleccionan las ganancias para controlar un sistema compuesto de una sola masa unitaria (es decir, sin fricción y sin rigidez).

La porción del control basada en el modelo aparece en una ley de control de la forma

$$f = \alpha f' + \beta, \tag{9.41}$$

en donde α y β son funciones o constantes y se seleccionan de manera que si f' se toma como la *nueva entrada* al sistema, *que parece ser una masa unitaria*. Con esta estructura de la ley de control, la ecuación del sistema (el resultado de combinar las ecuaciones (9.40) y (9.41)) es

$$m\ddot{x} + b\dot{x} + kx = \alpha f' + \beta. \tag{9.42}$$

Evidentemente, para que el sistema aparezca como una masa unitaria desde la entrada f', para este sistema específico debemos seleccionar α y β de la siguiente manera:

$$\alpha = m,$$
$$\beta = b\dot{x} + kx. \tag{9.43}$$

Al hacer estas asignaciones y sustituir en la ecuación (9.42), tenemos el sistema

$$\ddot{x} = f'. \tag{9.44}$$

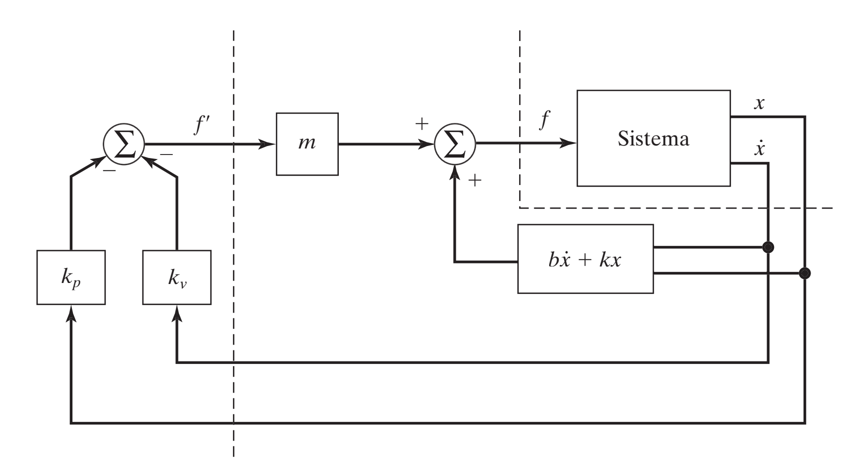


FIGURA 9.8: Un sistema de control de lazo cerrado que emplea el método de control particionado.

Ésta es la ecuación de movimiento para una masa unitaria. Ahora procederemos como si la ecuación (9.44) fuera la dinámica de lazo abierto de un sistema a controlar. Diseñamos una ley de control para calcular f', igual que antes:

$$f' = -k_v\dot{x} - k_p x. \tag{9.45}$$

Al combinar esta ley de control con la ecuación (9.44) se produce

$$\ddot{x} + k_v\dot{x} + k_p x = 0. \tag{9.46}$$

Bajo esta metodología es fácil establecer las ganancias de control, además de que es algo independiente de los parámetros del sistema; esto es,

$$k_v = 2\sqrt{k_p} \tag{9.47}$$

debe ser válida para el amortiguamiento crítico. La figura 9.8 muestra un diagrama de bloques del controlador particionado que se usó para controlar el sistema de la figura 9.6.

EJEMPLO 9.5

Si los parámetros del sistema de la figura 9.6 son $m = 1$, $b = 1$ y $k = 1$, encuentre α, β y las ganancias k_p y k_v para una ley de control de regulación de posición que haga que el sistema esté críticamente amortiguado, con una rigidez de lazo cerrado de 16.0.

Elegimos

$$\begin{aligned} \alpha &= 1, \\ \beta &= \dot{x} + x, \end{aligned} \tag{9.48}$$

tal que el sistema aparezca como una masa unitaria para la entrada ficticia f'. Luego asignamos a la ganancia k_p la rigidez de lazo cerrado deseada y hacemos que $k_v = 2\sqrt{k_p}$ para un amortiguamiento crítico.

Esto nos da

$$k_p = 16.0,$$
$$k_v = 8.0. \quad (9.49)$$

9.6 CONTROL DE SEGUIMIENTO DE TRAYECTORIAS

Vamos a mejorar nuestro controlador de manera que pueda hacer que el bloque siga una trayectoria en lugar de mantenerlo en una sola ubicación. Esta trayectoria está dada por una función del tiempo, $x_d(t)$, la cual especifica la posición deseada del bloque. Suponemos que la trayectoria es uniforme (es decir, que las primeras dos derivadas existen) y que nuestro generador de trayectorias proporciona x_d, $\dot{x}_d$ y $\ddot{x}_d$ en todos los tiempos t. Definiremos el error de servo entre las trayectorias deseada y real como $e = x_d - x$. Una ley de servocontrol que producirá el seguimiento de la trayectoria es

$$f' = \ddot{x}_d + k_v\dot{e} + k_p e. \quad (9.50)$$

Podemos ver que la ecuación (9.50) es una buena opción si la combinamos con la de movimiento de una masa unitaria (9.44), la cual nos lleva a

$$\ddot{x} = \ddot{x}_d + k_v\dot{e} + k_p e, \quad (9.51)$$

o a

$$\ddot{e} + k_v\dot{e} + k_p e = 0. \quad (9.52)$$

Ésta es una ecuación diferencial de segundo orden para la cual podemos elegir los coeficientes, de manera que podemos diseñar cualquier respuesta. (A menudo, la elección es amortiguamiento crítico). Algunas veces se dice que dicha ecuación está escrita en **espacio de error**, ya que describe la evolución de los errores en relación a la trayectoria deseada. La figura 9.9 muestra un diagrama de bloques de nuestro controlador de seguimiento de trayectorias.

Si nuestro modelo es perfecto (es decir, conocemos de m, b y k) y si no hay ruido ni error inicial, el bloque seguirá con exactitud la trayectoria deseada. Si hay un error inicial, éste se suprimirá de acuerdo con la ecuación (9.52), y de ahí en adelante, el sistema seguirá la trayectoria con exactitud.

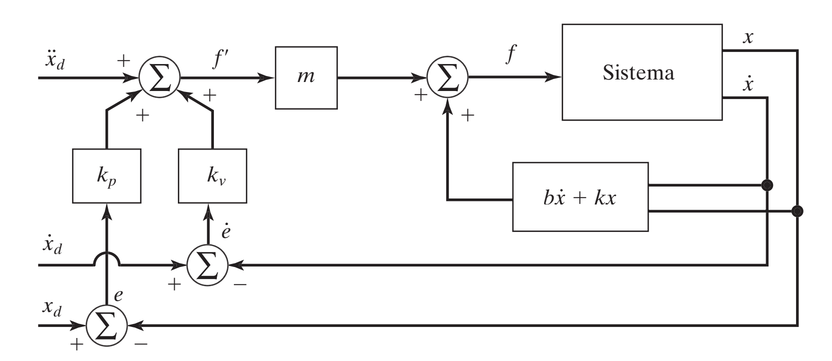


FIGURA 9.9: Un controlador de seguimiento de trayectoria para el sistema de la figura 9.6.

9.7 RECHAZO DE PERTURBACIONES

Uno de los propósitos de un sistema de control es proporcionar un **rechazo de perturbaciones**, esto es, para mantener un buen rendimiento (es decir, minimizar errores) aun en presencia de ciertos disturbios externos o **ruido**. En la figura 9.10 mostraremos el controlador de seguimiento de trayectorias con una entrada adicional: una fuerza perturbadora f_{dist}. Un análisis de nuestro sistema de lazo cerrado nos lleva a la ecuación de error:

$$\ddot{e} + k_v\dot{e} + k_p e = f_{\text{dist}}. \tag{9.53}$$

La (9.53) es una ecuación diferencial controlada por una función de fuerza. Si se sabe que f_{dist} está **delimitada**; es decir, que existe una constante a tal que

$$\max_t f_{\text{dist}}(t) < a, \tag{9.54}$$

entonces la solución de la ecuación diferencial $e(t)$ también está delimitada. Este resultado se debe a una propiedad de los sistemas lineales estables conocida como estabilidad de **entrada delimitada, salida delimitada** o **BIBO** [3, 4]. Este resultado bastante básico asegura que, para una clase extensa de posibles perturbaciones, podemos por lo menos estar seguros que el sistema permanecerá estable.

Error de estado estable

Consideremos el tipo de perturbación más simple; a saber, que f_{dist} sea una constante. En este caso podemos realizar un **análisis de estado estable** con el sistema en su estado inactivo (es decir, cuando las derivadas de todas las variables del sistema son iguales a cero). Si hacemos las derivadas iguales a cero en la ecuación (9.53) se produce la siguiente ecuación de estado estable:

$$k_p e = f_{\text{dist}}, \tag{9.55}$$

o

$$e = f_{\text{dist}}/k_p. \tag{9.56}$$

El valor de e dado por la ecuación (9.56) representa un **error de estado estable**. Por ende, queda claro que entre mayor sea la ganancia de posición k_p, menor será el error de estado estable.

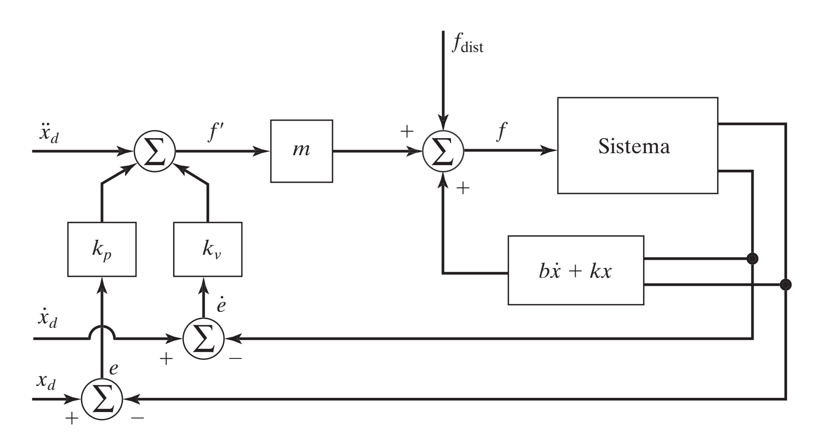


FIGURA 9.10: Un sistema de control de seguimiento de trayectorias en donde actúa una perturbación.

Adición de un término integral

Para eliminar el error de estado estable, algunas veces se utiliza una ley de control modificada. La modificación implica agregar un término *integral* a la ley de control, la cual se convierte en

$$f' = \ddot{x}_d + k_v\ddot{e} + k_p e + k_i \int e dt, \tag{9.57}$$

lo que produce la siguiente ecuación de error:

$$\ddot{e} + k_v\dot{e} + k_p e + k_i \int e dt = f_{\text{dist}}. \tag{9.58}$$

El término se agrega para que el sistema no tenga un error de estado estable en presencia de perturbaciones constantes. Si $e(t) = 0$ para $t < 0$, podemos escribir la ecuación (9.58) para $t > 0$ como

$$\dddot{e} + k_v\ddot{e} + k_p\dot{e} + k_i e = \dot{f}_{\text{dist}}, \tag{9.59}$$

la que, en el estado estable (para una perturbación constante) se convierte en

$$k_i e = 0, \tag{9.60}$$

de manera que

$$e = 0. \tag{9.61}$$

Con esta ley de control, el sistema se convierte en un sistema de tercer orden y se puede resolver la ecuación diferencial correspondiente (de tercer orden) para encontrar la respuesta del sistema a las condiciones iniciales. A menudo, k_i se mantiene considerablemente pequeña de manera que el sistema de tercer orden esté "cerca" del sistema de segundo orden sin este término (es decir, que puede realizarse un análisis de polos dominantes). La forma de la ley de control (9.57) se llama **ley de control PID**, o ley de control "proporcional, integral, derivada" [4]. Por simplicidad, en las leyes de control que desarrollamos en este libro las ecuaciones generalmente no muestran un término integral.

9.8 COMPARACIÓN ENTRE CONTROL DE TIEMPO CONTINUO Y DISCRETO

En los sistemas de control que hemos visto hemos supuesto implícitamente que la computadora de control realiza el cálculo de la ley de control en tiempo cero (es decir, infinitamente rápido), de manera que el valor de la fuerza accionadora f es una función continua de tiempo. Desde luego que en la realidad este cálculo requiere de cierto tiempo, y la fuerza comandada resultante es por lo tanto una función discreta de "escalera". Emplearemos esta aproximación de una computadora de control muy veloz a lo largo del libro. Esta aproximación es aceptable si la velocidad a la que se calculan los nuevos valores de f es mucho más rápida que la frecuencia natural del sistema que se está controlando. Al analizar el sistema [3] en el campo de **control de tiempo discreto** o **control digital** no se hace esta aproximación, sino que se toma en cuenta la **velocidad de servo** del sistema de control.

De manera general supondremos que los cálculos pueden realizarse lo suficientemente rápido como para que nuestra suposición de tiempo continuo sea válida. Aquí surge una pregunta: ¿Qué tan rápido es suficiente? Hay varios puntos que necesitamos tener en cuenta al elegir una velocidad de servo (o muestra) lo suficientemente rápida:

Rastreo de entradas de referencia: El contenido de frecuencia de la entrada deseada o de referencia impone un límite inferior absoluto en la frecuencia de muestreo, que debe ser por lo menos el doble del ancho de banda de las entradas de referencia. Por lo general este factor no es limitante.

Rechazo de perturbaciones: En el rechazo de perturbaciones se proporciona un límite superior en el rendimiento mediante un sistema de tiempo continuo. Si el periodo de muestreo es mayor que el tiempo de correlación de los efectos de la perturbación (suponiendo un modelo estadístico para disturbios aleatorios), entonces estos disturbios no se suprimirán. Una buena regla empírica es que el periodo de muestreo debe ser 10 veces más corto que el tiempo de correlación del ruido [3].

Control de alias: Siempre que se utilice un sensor análogo en un esquema de control digital habrá un problema con el efecto alias (aquél que produce la confusión de diferentes señales continuas cuando se les muestrea a una frecuencia demasiado baja), a menos que la salida del sensor esté estrictamente limitada en banda. En la mayoría de los casos, los sensores no tienen una salida limitada en banda, por lo que debe elegirse la proporción de muestreo de manera que la cantidad de energía que aparezca en la señal con alias sea pequeña.

Resonancias estructurales: En nuestra caracterización de la dinámica de un manipulador no hemos incluido modos de doblamiento. Todos los mecanismos reales tienen una rigidez finita y están sujetos a varios tipos de vibraciones. Si es importante suprimir estas vibraciones (y a menudo lo es), debemos elegir una velocidad de muestreo de cuando menos el doble de la frecuencia natural de estas resonancias. (Regresaremos al tema de la resonancia más adelante en este capítulo).

9.9 MODELADO Y CONTROL DE UNA SOLA ARTICULACIÓN

En esta sección desarrollaremos un modelo simplificado de una articulación giratoria individual de un manipulador. Haremos algunas suposiciones que nos permitirán modelar el sistema resultante como un sistema lineal de segundo orden. Para un modelo más completo de una articulación accionada, vea la referencia [5].

Un actuador común que se encuentra en muchos robots industriales es el motor de par de torsión de corriente directa (CD) (como el de la figura 8.18). La parte no giratoria del motor (el **estator**) consiste de una cubierta, varios cojinetes e imanes permanentes o bien electroimanes. Estos imanes del estator establecen un campo magnético sobre la parte giratoria del motor (el **rotor**). El rotor consiste de un eje y devanados a través de los cuales pasa la corriente para energizar el motor. La corriente se conduce hacia los devanados mediante escobillas, las cuales hacen contacto con el conmutador. El conmutador está cableado a los diversos devanados (también conocidos como **armadura**) de tal forma que siempre se produzca un momento de torsión en la dirección deseada. El fenómeno físico subyacente [6] que hace que un motor genere un momento de torsión cuando la corriente pasa a través de los devanados puede expresarse como

$$F = qV \times B, \tag{9.62}$$

en donde la carga q, que se desplaza con una velocidad V a través de un campo magnético B, experimenta una fuerza F. Las cargas son las de los electrones que se mueven a través de los devanados, y el campo magnético es el que establecen los imanes del estator. Generalmente, la habilidad de un motor para producir momentos de torsión se define mediante una **constante de par de motor** individual, la cual relaciona la corriente de la armadura con el momento de torsión de salida como

$$\tau_m = k_m i_a. \tag{9.63}$$

Cuando un motor está girando actúa como un generador, y se desarrolla un voltaje sobre la armadura. Una segunda constante del motor, llamada **constante fem inversa**,[3] describe el voltaje generado para una velocidad de rotación dada:

$$v = k_e \dot{\theta}_m. \tag{9.64}$$

El hecho de que el conmutador esté cambiando la corriente a través de varios conjuntos de devanados generalmente hace que el momento de torsión producido contenga cierto **rizo de momento de torsión**. Aunque algunas veces es importante, por lo general este efecto puede ignorarse. (En cualquier caso es muy difícil de modelar; además que es complicado hacer una compensación del mismo, aún si se modela).

Inductancia de la armadura del motor

La figura 9.11 muestra el circuito eléctrico de la armadura. Los principales componentes son una fuente de voltaje (v_a), la inductancia de los devanados de la armadura (l_a), la resistencia de los devanados de la armadura (r_a) y la fem inversa generada, v. El circuito está descrito por una ecuación diferencial de primer orden:

$$l_a \dot{i}_a + r_a i_a = v_a - k_e \dot{\theta}_m. \tag{9.65}$$

Por lo general es deseable controlar el momento de torsión generado por el motor (en vez de la velocidad) con circuitos electrónicos de accionamiento. Estos circuitos de control detectan la corriente que pasa a través de la armadura y ajustan continuamente la fuente de voltaje v_a de manera que fluya una corriente i_a deseada a través de la armadura. Dicho circuito se conoce como controlador de motor **amplificador de corriente** [7]. En estos sistemas de control de corriente, la velocidad a la que la corriente de la armadura puede hacerse cambiar está limitada por la inductancia del motor l_a y por un límite superior en la capacidad de voltaje de la fuente de voltaje v_a. El efecto neto es el de un **filtro pasabajos** entre la corriente solicitada y el momento de torsión de salida.

Nuestra primera suposición simplificada es que la inductancia del motor puede ignorarse. Ésta es una suposición razonable cuando la frecuencia natural del sistema de control de lazo cerrado es considerablemente baja en comparación con la frecuencia de corte del filtro pasabajos implícito en los circuitos de control de corriente (debido a la inductancia). Esta suposición, junto con la de que el rizo del momento de torsión es un efecto que puede ignorarse, significa que podemos en principio comandar el momento de torsión de manera directa. Aunque tal vez tuviéramos que lidiar con un factor de escala (como k_m), supondremos que el actuador actúa como una fuente pura de momento de torsión que podemos comandar de manera directa.

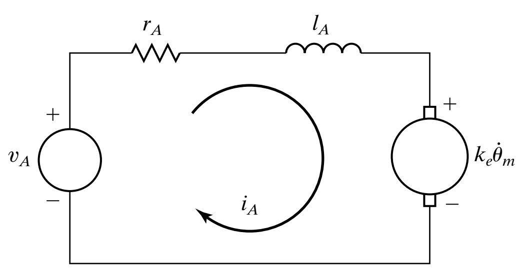


FIGURA 9.11: El circuito de la armadura de un motor de par de torsión de CD.

[3] "fem" significa fuerza electromotriz.

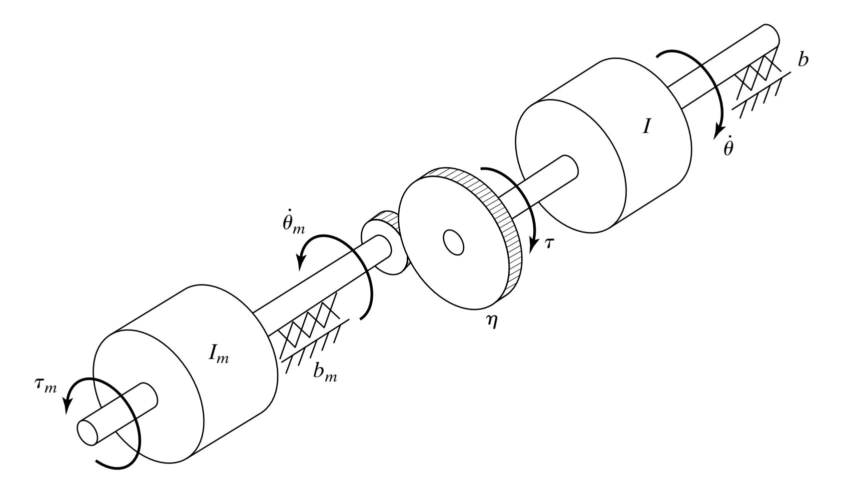


FIGURA 9.12: Modelo mecánico de un motor de par de torsión de CD conectado por medio de engranajes a una carga inercial.

Inercia efectiva

La figura 9.12 muestra el modelo mecánico del rotor de un motor de par de torsión de CD conectado a través de un engranaje a una carga inercial. El momento de torsión aplicado al rotor, τ_m, está dado por la ecuación (9.63) como una función de la corriente i_a que fluye en el circuito de la armadura. La razón de engranaje (η) produce un incremento en el momento de torsión visto en la carga, y una reducción en la velocidad de la carga, dados por

$$\begin{aligned} \tau &= \eta\tau_m, \\ \dot{\theta} &= (1/\eta)\dot{\theta}_m, \end{aligned} \tag{9.66}$$

en donde $\eta > 1$. Si escribimos un balance de momentos de torsión para este sistema en términos del momento de torsión en el rotor se produce la siguiente ecuación:

$$\tau_m = I_m\ddot{\theta}_m + b_m\dot{\theta}_m + (1/\eta)\left(I\ddot{\theta} + b\dot{\theta}\right), \tag{9.67}$$

en donde I_m e I son las inercias del rotor del motor y de la carga, respectivamente, y b_m y b son los coeficientes de fricción viscosa para el rotor y los cojinetes de la carga, respectivamente. Utilizando las relaciones (9.66) podemos escribir la ecuación (9.67) en términos de variables del motor:

$$\tau_m = \left(I_m + \frac{I}{\eta^2}\right)\ddot{\theta}_m + \left(b_m + \frac{b}{\eta^2}\right)\dot{\theta}_m \tag{9.68}$$

o en términos de variables de carga:

$$\tau = (I + \eta^2 I_m)\ddot{\theta} + (b + \eta^2 b_m)\dot{\theta}. \tag{9.69}$$

El término $I + \eta^2 I_m$ algunas veces se conoce como la **inercia efectiva** “vista” en la salida (lado del vínculo) del engranaje. De igual forma, el término $b + \eta^2 b_m$ puede llamarse **amortiguamiento efectivo**. Observe que en una articulación con alta razón de engranaje (es decir, que $\eta \gg 1$), la inercia del rotor del motor puede ser una porción considerable de la inercia efectiva combinada. Este efecto es el que nos permite suponer

que la inercia efectiva es una constante. Sabemos, del capítulo 6, que la inercia I de una articulación del mecanismo varía con la configuración y la carga. No obstante, en robots con altas razones de engranaje las variaciones representan un porcentaje más pequeño que en un manipulador de **control directo** (es decir, cuando $\eta = 1$). Para asegurar que el movimiento del vínculo del robot nunca esté subamortiguado, el valor utilizado para I debe ser el máximo del rango de valores que reciba I; a este valor le llamaremos $I_{máx}$. Esta elección produce un sistema que está críticamente amortiguado o sobreamortiguado en todas las situaciones. En el capítulo 10 trataremos la inercia variante directamente y no tendremos que hacer esta suposición.

EJEMPLO 9.6

Si la inercia aparente del vínculo, I, varía entre 2 y 6 Kg-m^2, la inercia del rotor es $I_m = 0.01$ y la proporción de engranajes es $\eta = 30$, ¿cuáles son los valores mínimo y máximo de la inercia efectiva?

La inercia efectiva mínima es

$$I_{\text{mín}} + \eta^2 I_m = 2.0 + (900)(0.01) = 11.0; \tag{9.70}$$

la máxima es

$$I_{\text{máx}} + \eta^2 I_m = 6.0 + (900)(0.01) = 15.0. \tag{9.71}$$

Por ende, podemos ver que, como un porcentaje de la inercia efectiva total, la variación de la inercia se reduce mediante el engranaje.

Flexibilidad no modelada

La otra suposición importante que hemos hecho en nuestro modelo es que el engranaje, los ejes, los cojinetes y el vínculo controlado no son flexibles. En realidad todos estos elementos tienen una rigidez finita, y si se modelara su flexibilidad se incrementaría el orden del sistema. El argumento para ignorar los efectos de la flexibilidad es que, si el sistema es lo suficientemente rígido, las frecuencias naturales de estas **resonancias no modeladas** son muy altas y pueden ignorarse en comparación con la influencia de los polos dominantes de segundo orden que hemos modelado.[4] El término "no modelado" se refiere al hecho de que, para los fines del análisis y diseño de sistemas de control, ignoramos estos efectos y utilizamos un modelo dinámico más simple, tal como la ecuación (9.69).

Como hemos optado por no modelar las flexibilidades estructurales en el sistema, debemos tener cuidado de no excitar estas resonancias. Una regla empírica [8] es que si la resonancia estructural más baja es ω_{res}, entonces debemos limitar nuestra frecuencia natural de lazo cerrado de acuerdo con

$$\omega_n \leq \frac{1}{2}\omega_{\text{res}}. \tag{9.72}$$

Esto proporciona cierta guía sobre cómo seleccionar las ganancias en nuestro controlador. Hemos visto que al incrementar las ganancias se produce una respuesta más rápida y un error de estado estable más bajo, pero ahora vemos que las resonancias estructurales no modeladas limitan la magnitud de las ganancias. Los manipuladores industriales comunes tienen resonancias estructurales en el rango de 5 a 25 Hz [8]. Los

[4] Éste es básicamente el mismo argumento que utilizamos para ignorar el polo debido a la inductancia del motor. Si lo hubiéramos incluido también se hubiera elevado el orden del sistema en general.

diseños recientes que utilizan arreglos de accionamiento directo y que no contienen flexibilidad introducida por sistemas de reducción y de transmisión tienen sus resonancias estructurales más bajas hasta en 70 Hz [9].

EJEMPLO 9.7

Considere el sistema de la figura 9.7 con los valores de parámetros $m = 1$, $b = 1$ y $k = 1$. Además se sabe que la resonancia no modelada más baja del sistema se encuentra en 8 radianes/segundo. Encuentre α, β y las ganancias k_p y k_v para una ley de control de posición de forma que el sistema esté críticamente amortiguado, que no excite la dinámica no modelada y que tenga la rigidez de lazo cerrado más alta posible.

Elegiremos

$$\alpha = 1,$$
$$\beta = \dot{x} + x, \tag{9.73}$$

de manera que el sistema aparezca como una masa unitaria de la entrada ficticia f'. Utilizando nuestra regla empírica (9.72), elegimos la frecuencia natural de lazo cerrado para que sea $\omega_n = 4$ radianes/segundo. De las ecuaciones (9.18) y (9.46) tenemos que $k_p = \omega_n^2$, por lo que

$$k_p = 16.0,$$
$$k_v = 8.0. \tag{9.74}$$

Estimación de la frecuencia resonante

Las mismas fuentes de flexibilidad estructural que vimos en el capítulo 8 dan origen a las resonancias. En cada caso en el que pueda identificarse una flexibilidad estructural es posible hacer un análisis aproximado de la vibración resultante si podemos describir la masa o inercia efectiva del miembro flexible. Esto se hace aproximando la situación mediante un sistema simple de resorte-masa que, según la ecuación (9.20), exhibe la frecuencia natural

$$\omega_n = \sqrt{k/m}, \tag{9.75}$$

en donde k es la rigidez del miembro flexible y m es la masa equivalente desplazada en las vibraciones.

EJEMPLO 9.8

Un eje (se supone sin masa) con una rigidez de 400 Nt-m/radián controla una inercia giratoria de 1 Kg-m^2. Si la rigidez del eje se ignoró al modelar la dinámica, ¿cuál es la frecuencia de esta resonancia no modelada?

Utilizando la ecuación (9.75) tenemos que

$$\omega_{res} = \sqrt{400/1} = 20 \text{ rad/segundos} = 20/(2\pi)\text{Hz} \cong 3.2 \text{ Hz}. \tag{9.76}$$

Para los fines de una estimación aproximada de la frecuencia resonante más baja de vigas y ejes, la referencia [10] sugiere utilizar un **modelo concentrado** de la masa. Ya tenemos fórmulas para estimar la rigidez en los extremos de las vigas y los ejes; estos modelos concentrados proporcionan la masa o inercia efectiva necesaria para nuestra

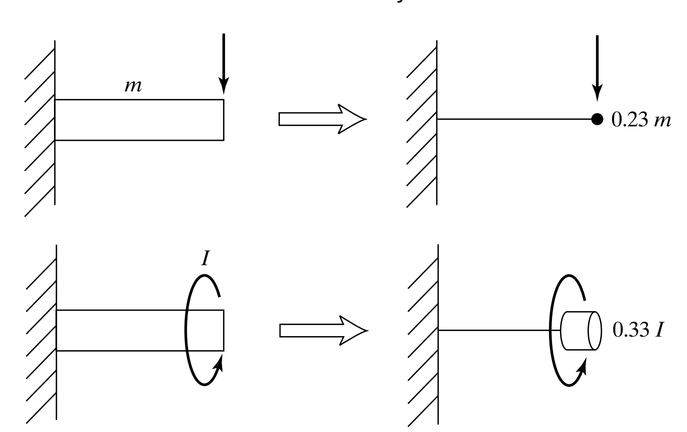


FIGURA 9.13: Modelos concentrados de vigas para la estimación de las resonancias lateral y torsional más baja.

estimación de la frecuencia resonante. La figura 9.13 muestra los resultados de un análisis de energía [10] que sugiere que una viga de masa m sea reemplazada por una masa puntual en el extremo de 0.23 m y, de igual forma, que una inercia distribuida de I sea reemplazada por una inercia concentrada de 0.33 I al final del eje.

EJEMPLO 9.9

Un vínculo con una masa de 4.347 Kg tiene una rigidez lateral en su punto más extremo de 3600 Nt/m. Suponiendo que el sistema de control sea completamente rígido, la resonancia debida a la flexibilidad del vínculo limitará las ganancias de control. ¿Cuál es el valor de ω_{res}?

La masa de 4.347 Kg está distribuida a lo largo del vínculo. Utilizando el método de la figura 9.13, la masa efectiva es (0.23)(4.347) ≅ 1.0 Kg. Por lo tanto, la frecuencia de vibración es

$$\omega_{\text{res}} = \sqrt{3600/1.0} = 60 \text{ radianes/segundos} = 60/(2\pi)\text{Hz} \cong 9.6 \text{ Hz}. \tag{9.77}$$

Hay que incluir las flexibilidades estructurales en el modelo del sistema utilizado para la síntesis de la ley de control si deseamos alcanzar anchos de banda de lazo cerrado mayores que los proporcionados por la ecuación (9.75). Los modelos de sistema resultantes son de mayor orden, y las técnicas de control aplicables a esta situación se vuelven bastante sofisticadas. Dichos esquemas de control se encuentran actualmente más allá de lo más avanzado de la práctica industrial, pero son un área activa de investigación [11, 12].

Control de una sola articulación

En resumen, haremos las tres siguientes suposiciones principales:

1. La inductancia del motor l_a puede ignorarse.
2. Tomando en cuenta una alta razón de engranaje, modelamos la inercia efectiva como una constante igual a $I_{\text{máx}} + \eta^2 I_m$.
3. Las flexibilidades estructurales se ignoran, exceptuando que la resonancia estructural más baja ω_{res} se utiliza para establecer las ganancias del servo.

Con estas suposiciones, una sola articulación de un manipulador puede controlarse con el controlador particionado dado por

$$\alpha = I_{\text{máx}} + \eta^2 I_m,$$

$$\beta = (b + \eta^2 b_m)\dot{\theta}, \tag{9.78}$$

$$\tau' = \ddot{\theta}_d + k_v \dot{e} + k_p e. \tag{9.79}$$

La dinámica de lazo cerrado del sistema resultante es

$$\ddot{e} + k_v \dot{e} + k_p e = \tau_{\text{dist}}, \tag{9.80}$$

en donde las ganancias se seleccionan como

$$k_p = \omega_n^2 = \frac{1}{4}\omega_{\text{res}}^2,$$

$$k_v = 2\sqrt{k_p} = \omega_{\text{res}}. \tag{9.81}$$

9.10 ARQUITECTURA DE UN CONTROLADOR DE ROBOT INDUSTRIAL

En esta sección veremos brevemente la arquitectura del sistema de control del robot industrial Unimation PUMA 560. Como se muestra en la figura 9.14, la arquitectura de hardware es la de una jerarquía de dos niveles, con una computadora DEC LSI-11 que sirve como la computadora de control "maestro" de nivel superior, la cual pasa los comandos a seis microprocesadores Rockwell 6503.[5] Cada uno de estos microprocesadores controla una articulación individual con una ley de control PID parecida a la que presentamos en este capítulo. Cada articulación del PUMA 560 está equipada con un codificador óptico incremental. Los codificadores tienen una interfaz con un contador arriba/abajo, que puede ser leído por el microprocesador para obtener la posición actual de la articulación. No hay tacómetros en el PUMA 560; las posiciones de las articulaciones se diferencian en ciclos de servo subsecuentes para obtener una estimación de la velocidad de las articulaciones. Para comandar momentos de torsión a los motores de par de torsión de CD, el microprocesador se conecta a un convertidor digital-análogo (D/A) para que se puedan enviar las corrientes del motor a los circuitos

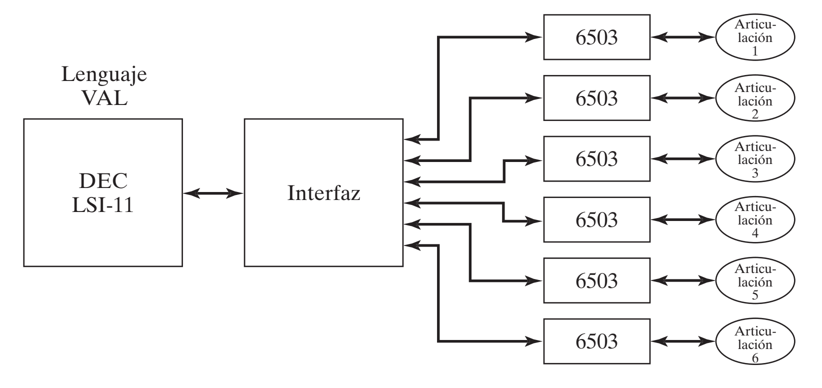


FIGURA 9.14: Arquitectura computacional jerárquica del sistema de control del robot PUMA 560.

[5] Estas computadoras simples de 8 bits son ya una tecnología obsoleta. Actualmente los controladores de robots están basados en microprocesadores de 32 bits.

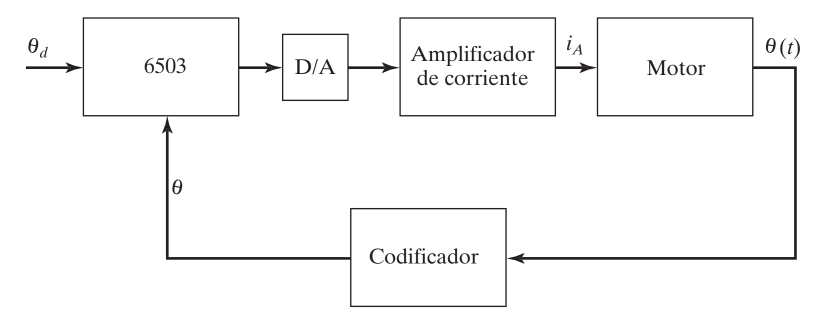


FIGURA 9.15: Bloques funcionales del sistema de control de articulaciones del PUMA 560.

controladores de corriente. La corriente que fluye a través del motor se controla en un circuito análogo ajustando el voltaje a través de la armadura según sea necesario para mantener la corriente deseada. La figura 9.15 muestra un diagrama de bloques.

Cada 28 milisegundos la computadora LSI-11 envía un nuevo comando de posición (**punto de ajuste**) a los microprocesadores de las articulaciones. Estos microprocesadores están funcionando en un ciclo de 0.875 milisegundos. Durante este tiempo interpolan el punto de ajuste de la posición deseada, calculan el error de servo, calculan la ley de control PID y ordenan un nuevo valor de momento de torsión a los motores.

La computadora LSI-11 se encarga de todas las operaciones de "alto nivel" del sistema de control en general. Primero que nada se encarga de interpretar uno por uno los comandos del programa en VAL (lenguaje de programación de robots de Unimation). Cuando se interpreta un comando de movimiento, la computadora LSI-11 debe realizar todos los cálculos necesarios de cinemática inversa, planear una trayectoria deseada y empezar a generar puntos vía de la trayectoria cada 28 milisegundos para los controladores de las articulaciones.

La computadora LSI-11 también está conectada con periféricos estándar tales como la terminal y una unidad de disco flexible. Además está conectada a un **dispositivo de enseñanza**, que es una caja portátil de botones que permite al operador mover el robot en varios modos. Por ejemplo, el sistema del PUMA 560 permite al usuario mover el robot de manera incremental, en las coordenadas de las articulaciones o en las coordenadas cartesianas desde el dispositivo de enseñanza. En este modo, los botones del dispositivo de enseñanza hacen que se calcule una trayectoria "al instante", la cual se pasa a los microprocesadores de control de las articulaciones.

BIBLIOGRAFÍA

[1] W. Boyce y R. DiPrima, *Elementary Differential Equations*, 3a. edición, John Wiley and Sons, Nueva York, 1977.

[2] E. Purcell, *Calculus with Analytic Geometry*, Meredith Corporation, Nueva York, 1972.

[3] G. Franklin y J. D. Powell, *Digital Control of Dynamic Systems*, Addison-Wesley, Reading, MA, 1980.

[4] G. Franklin, J. D. Powell y A. Emami-Naeini, *Feedback Control of Dynamic Systems*, Addison-Wesley, Reading, MA, 1986.

[5] J. Luh, "Conventional Controller Design for Industrial Robots – a Tutorial", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. *SMC-13*, Núm. 3, junio de 1983.

[6] D. Halliday y R. Resnik, *Fundamentals of Physics*, Wiley, Nueva York, 1970.

[7] Y. Koren y A. Ulsoy, "Control of DC Servo-Motor Driven Robots", *Proceedings of Robots 6 Conference*, SME, Detroit, marzo de 1982.

[8] R. P. Paul, *Robot Manipulators*, MIT Press, Cambridge, MA, 1981.

[9] H. Asada y K. Youcef-Toumi, *Direct-Drive Robots – Theory and Practice,* MIT Press, Cambridge, MA, 1987.

[10] J. Shigley, *Mechanical Engineering Design*, 3a. edición, McGraw-Hill, Nueva York, 1977.

[11] W. Book, "Recursive Lagrangian Dynamics of Flexible Manipulator Arms", *The International Journal of Robotics Research*, Vol. 3, Núm. 3, 1984.

[12] R. Cannon y E. Schmitz, "Initial Experiments on the End-Point Control of a Flexible One Link Robot", *The International Journal of Robotics Research*, Vol. 3, Núm. 3, 1984.

[13] R. J. Nyzen, *Analysis and Control of an Eight-Degree-of-Freedom Manipulator*, tesis de maestría de la Universidad de Ohio, ingeniería mecánica, Dr. Robert L. Williams II, asesor, agosto de 1999.

[14] R. L. Williams II, *Local Performance Optimization for a Class of Redundant Eight-Degree-of-Freedom Manipulators*, NASA Technical Paper 3417, NASA Langley Research Center, Hampton, VA, marzo de 1994.

EJERCICIOS

9.1 [20] Para una ecuación diferencial de segundo orden con raíces complejas

$$s_1 = \lambda + \mu i,$$
$$s_2 = \lambda - \mu i,$$

demuestre que la solución general

$$x(t) = c_1 e^{s_1 t} + c_2 e^{s_2 t},$$

puede escribirse así:

$$x(t) = c_1 e^{\lambda t} \cos(\mu t) + c_2 e^{\lambda t} \operatorname{sen}(\mu t).$$

9.2 [13] Calcule el movimiento del sistema de la figura 9.2 si los valores de los parámetros son $m = 2$, $b = 6$ y $k = 4$, y si el bloque (inicialmente en reposo) se libera desde la posición $x = 1$.

9.3 [13] Calcule el movimiento del sistema de la figura 9.2 si los valores de los parámetros son $m = 1$, $b = 2$ y $k = 1$, y si el bloque (inicialmente en reposo) se libera desde la posición $x = 4$.

9.4 [13] Calcule el movimiento del sistema de la figura 9.2 si los valores de los parámetros son $m = 1$, $b = 4$ y $k = 5$, y si el bloque (inicialmente en reposo) se libera desde la posición $x = 2$.

9.5 [15] Calcule el movimiento del sistema de la figura 9.2 si los valores de los parámetros son $m = 1$, $b = 7$ y $k = 10$, y si el bloque se libera desde la posición $x = 1$ con una velocidad inicial de $v = 2$.

9.6 [15] Use el elemento (1, 1) de la ecuación (6.60) para calcular la variación (como un porcentaje del valor máximo) de la inercia "vista" por la articulación 1 de este robot a medida que cambia de configuración. Use los valores numéricos

$$l_1 = l_2 = 0.5 \text{ m},$$
$$m_1 = 4.0 \text{ Kg},$$
$$m_2 = 2.0 \text{ Kg}.$$

Considere que el robot es de accionamiento directo y que la inercia del rotor puede ignorarse.

9.7 [17] Repita el ejercicio 9.6 para el caso de un robot con engranajes (use $\eta = 20$) y una inercia del rotor de $I_m = 0.01$ Kg-m^2.

9.8 [18] Considere el sistema de la figura 9.6 con los valores de parámetros $m = 1$, $b = 4$ y $k = 5$. También se sabe que el sistema posee una resonancia no modelada a $\omega_{res} =$ 6.0 radianes/segundo. Determine las ganancias k_v y k_p que amortigüen críticamente el sistema con una rigidez tan alta como sea razonable.

9.9 [25] En un sistema como el de la figura 9.12, la carga inercial I varía entre 4 y 5 Kg-m^2. La inercia del rotor es $I_m = 0.01$ Kg-m^2 y la proporción de engranajes es $\eta = 10$. El sistema posee resonancias no modeladas a 8.0, 12.0 y 20.0 radianes/segundo. Diseñe α y β del controlador particionado y proporcione los valores de k_p y k_v tales que el sistema nunca esté subamortiguado y nunca excite resonancias, pero que sea lo más rígido posible.

9.10 [18] Un diseñador de un robot de accionamiento directo sospecha que la resonancia debida a la flexibilidad de las vigas del vínculo en sí será la causa de la resonancia no modelada más baja. Si el vínculo es aproximadamente una viga de corte transversal cuadrado de dimensiones 5 × 5 × 50 cm, con grosor de pared de 1 cm y una masa total de 5 Kg, estime el valor de ω_{res}.

9.11 [15] El vínculo de un robot de accionamiento directo se controla a través de un eje con una rigidez de 1000 Nt-m/radián. La inercia del vínculo es de 1 Kg-m^2. Suponiendo que el eje no tenga masa, ¿cuál es el valor de ω_{res}?

9.12 [18] Un eje con una rigidez de 500 Nt-m/radián controla la entrada de un par de engranajes rígidos con $\eta = 8$. La salida de los engranajes controla un vínculo rígido con una inercia de 1 Kg-m^2. ¿Cuál es la ω_{res} ocasionada por la flexibilidad del eje?

9.13 [25] Un eje con una rigidez de 500 Nt-m/radián controla la entrada de un par de engranajes rígidos con $\eta = 8$. El eje tiene una inercia de 0.1 Kg-m^2. La salida de los engranajes controla un vínculo rígido con una inercia de 1 Kg-m^2. ¿Cuál es la ω_{res} ocasionada por la flexibilidad del eje?

9.14 [28] En un sistema como el de la figura 9.12, la carga inercial I varía entre 4 y 5 Kg-m^2. La inercia del rotor es $I_m = 0.01$ Kg-m^2 y la proporción de engranajes es $\eta = 10$. El sistema posee una resonancia no modelada debida a la rigidez de un punto del extremo del vínculo de 2400 Nt-m/radián. Diseñe α y β del controlador particionado y proporcione los valores de k_p y k_v tales que el sistema nunca esté subamortiguado y nunca excite resonancias, pero que sea lo más rígido posible.

9.15 [25] Un eje de acero con una longitud de 30 cm y un diámetro de 0.2 cm controla el engranaje de entrada de una reducción de $\eta = 8$. El engranaje de salida rígido controla un eje de acero con una longitud de 30 cm y un diámetro de 0.3 cm. ¿Cuál es el rango de frecuencias resonantes que se observan si la inercia de la carga varía entre 1 y 4 Kg-m^2?

EJERCICIO DE PROGRAMACIÓN (PARTE 9)

Deseamos simular un sistema de control simple de seguimiento de trayectorias para un brazo planar de tres vínculos. Este sistema de control se implementará como una ley de control PD (proporcional derivativo) de articulación independiente. Ajuste las ganancias del servo para lograr rigideces de lazo cerrado de 175.0, 110.0 y 20.0 para las articulaciones 1 a 3, respectivamente. Trate de lograr un amortiguamiento crítico aproximado.

Use la rutina de simulación **ACTUALIZA** para simular un servo de tiempo discreto ejecutándose a 100 Hz; esto es, calcule la ley de control a 100 Hz, no a la frecuencia del proceso de integración numérica. Pruebe el esquema de control con los siguientes ejercicios.

1. Inicie el brazo en $\Theta = (60, -110, 20)$ y ordénele quedarse ahí hasta que *tiempo* $= 3.0$, cuando los puntos de ajuste deban cambiar instantáneamente a $\Theta = (60, -50, 20)$. Esto es, proporcione una entrada de escalón de 60 grados para la articulación 2. Registre el historial de error–tiempo para cada articulación.

2. Controle el brazo para que siga la trayectoria de trazador cúbico del ejercicio de programación, parte 7. Registre el historial de error -tiempo para cada articulación.

EJERCICIO MATLAB 9

Este ejercicio se enfoca en la simulación del control de articulación independiente linealizado para la articulación tipo hombro (articulación 2) del brazo manipulador AAI ARMIII (Manipulador de Investigación Avanzada II) de ocho ejes de la NASA; vea la referencia [14]. Se supone una familiaridad con los sistemas de control de retroalimentación clásicos lineales, incluyendo los diagramas de bloque y las transformadas de Laplace. Utilizaremos Simulink, la interfaz gráfica de usuario de MATLAB.

La figura 9.16 muestra el modelo dinámico de un sistema de lazo abierto linealizado para la articulación/vínculo de hombro electromecánico del ARMII, accionada por un servomotor de armadura-controlador de CD. La entrada de lazo abierto es el voltaje de referencia V_{ref} (incrementado hasta el voltaje de la armadura mediante un amplificador), y la salida de interés es el ángulo del eje de carga, ThetaL. La figura también muestra el diagrama de control de retroalimentación, en donde el ángulo del eje de carga se detecta mediante un codificador óptico y se proporciona como retroalimentación al controlador PID. La tabla describe todos los parámetros y las variables del sistema.

Si reflejamos la inercia y el amortiguamiento del eje de carga al eje del motor, los coeficientes de inercia polar efectiva y de amortiguamiento son $J = J_M + J_L(t)/n^2$ y $C = C_M + C_L/n^2$. En virtud de la gran razón de engranaje n, estos valores efectivos no son muy distintos de los valores del eje del motor. Por lo tanto, la razón de engranaje nos permite ignorar las variaciones en la inercia del eje de carga dependiente de la configuración $J_L(t)$ y sólo establecer un valor promedio razonable.

Los parámetros constantes de la articulación tipo hombro del ARMII aparecen en la tabla que se muestra a continuación [13]. Observe que podemos usar las unidades inglesas directamente ya que su efecto se cancela dentro del diagrama de control. Además podemos usar directamente unidades de grados para el ángulo. Desarrolle un modelo en Simulink para simular el modelo de control de una articulación a partir del diagrama de control

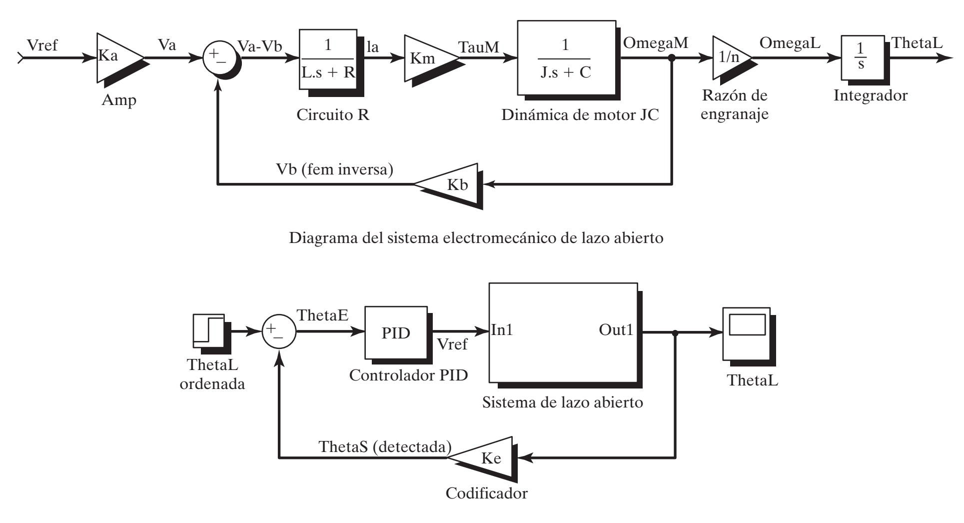


FIGURA 9.16: Modelo dinámico del sistema de lazo abierto linealizado para la articulación/vínculo de hombro electromecánico del ARMII, accionada por un servomotor de CD de armadura-controlador.

$V_a(t)$	Voltaje de la armadura	$\tau_M(t)$	Par de torsión de motor generado	$\tau_L(t)$	Momento de torsión de la carga
$L = 0.0006\ H$	Inductancia de la armadura	$\theta_M(t)$	Ángulo del eje del motor	$\theta_L(t)$	Ángulo del eje de carga
$R = 1.40\Omega$	Resistencia de la armadura	$\omega_M(t)$	Velocidad del eje del motor	$\omega_L(t)$	Velocidad del eje de carga
$i_a(t)$	Corriente de la armadura	$J_M = 0.00844$ lb_f-pulg-s^2	Inercia polar concentrada de motor	$J_L(t) = 1$ lb_f-pulg-s^2	Inercia polar concentrada de carga
$V_b(t)$	Voltaje de fem inversa	$C_M = 0.00013$ lb_f–pulg/grad/seg	Coeficiente de amortiguamiento viscoso del eje del motor	$C_L = 0.5$ lb_f–pulg/grad/seg	Coeficiente de amortiguamiento viscoso del eje de carga
$K_a = 12$	Ganancia del amplificador	$n = 200$	Razón de engranaje	$g = 0$ pulg/s^2	Gravedad (ignore la gravedad al principio)
$K_b = 0.00867$ V/grad/s	Constante de fem inversa	$K_M = 4.375$ lb_f–pulg/A	Constante de momento de torsión	$K_e = 1$	Función de transferencia de codificador

retroalimentado y del modelo que se muestra; use los parámetros específicos de la tabla. Para el caso nominal, determine las ganancias PID mediante prueba y error para un "buen" rendimiento (porcentaje razonable de sobredisparo, tiempo de subida, tiempo pico y tiempo de establecimiento). Simule el movimiento resultante para mover esta articulación tipo hombro para una entrada escalón de 0 a 60 grados. Trace el valor simulado del ángulo de carga sobre el tiempo, y también la velocidad angular del eje de carga sobre el tiempo. Además, trace el esfuerzo de control; esto es, el voltaje de armadura V_a sobre el tiempo. (En la misma gráfica, también proporcione la fem inversa V_b).

Ahora pruebe algunos cambios; es muy fácil y entretenido cambiar en Simulink:

1) La entrada en escalón es frustrante para el diseño de controladores, así que pruebe mejor una entrada en escalón tipo rampa: suba uniformemente de 0 a 60 grados en 1.5 segundos y después sostenga el comando de 60 grados durante todo el tiempo mayor de los 1.5 seg. Rediseñe las ganancias PID y vuelva a estimular.

2) Investigue si el inductor L es significativo en este sistema. (El sistema eléctrico se eleva mucho más rápido que el sistema mecánico; este efecto puede representarse mediante constantes de tiempo).

3) No tenemos una buena estimación para la inercia de carga y el amortiguamiento (J_L y C_L). Con sus mejores ganancias PID de antes, investigue qué tanto pueden crecer estos valores (escale los parámetros nominales hacia arriba por igual) antes de que afecten al sistema.

4) Ahora incluya el efecto de la gravedad como una perturbación para el par de motor T_M. Suponga que la masa del robot en movimiento es de 200 lb y que la longitud móvil más allá de la articulación 2 es de 6.4 pies. Pruebe buscando las ganancias PID "buenas" nominales que encontró; vuelva a diseñar si es necesario. La configuración en ceros del ángulo de carga θ_2 tipo hombro es directamente hacia arriba.

CAPÍTULO 10

Control no lineal de los manipuladores

10.1 INTRODUCCIÓN

En el capítulo 9 hicimos varias aproximaciones para permitir un análisis lineal del problema del control de manipuladores; la más importante fue que cada articulación podía considerarse independiente y que la inercia "vista" por cada actuador de articulación era constante. En implementaciones de controladores lineales, como los que se presentaron en el capítulo anterior, esta aproximación produce un amortiguamiento no uniforme a lo largo del espacio de trabajo y otros efectos indeseables. En este capítulo presentaremos una técnica de control más avanzada para la cual no habrá que hacer esta suposición.

En el capítulo 9 modelamos el manipulador mediante n ecuaciones diferenciales independientes de segundo orden y basamos nuestro controlador en ese modelo. En este capítulo basaremos nuestro diseño del controlador directamente en la ecuación diferencial del vector no lineal de $n \times 1$ del movimiento, derivada en el capítulo 6 para un manipulador general.

El campo de la teoría de control no lineal es extenso; por lo tanto debemos restringir nuestra atención a uno o dos métodos que parecen adaptarse bien a los manipuladores mecánicos. Por consecuencia, nuestro mayor énfasis aquí será sobre un método específico, que aparentemente se propuso por primera vez en la referencia [1] y al que se nombró **método de momento de torsión calculado** en [2, 3]. También presentaremos un método de análisis de estabilidad de sistemas no lineales, conocido como el método de **Lyapunov** [4].

Comenzaremos nuestro examen de las técnicas no lineales para controlar un manipulador, para lo cual regresamos a un sistema de fricción tipo masa–resorte muy simple, con un solo grado de libertad.

10.2 SISTEMAS NO LINEALES Y VARIANTES EN EL TIEMPO

En el desarrollo anterior tratamos con una ecuación diferencial lineal de coeficientes constantes. Esta forma matemática surgió porque el sistema de fricción tipo masa–resorte de la figura 9.6 se modeló como un sistema lineal invariante en el tiempo. Para sistemas cuyos parámetros varían en el tiempo, o sistemas que por naturaleza son no lineales, las soluciones son más difíciles.

Cuando las no linealidades no son severas, puede utilizarse la **linealización local** para derivar modelos lineales que son aproximaciones de las ecuaciones no lineales en ubicaciones adyacentes a un **punto de operación**. Desafortunadamente, el problema del control de manipuladores no se adapta bien a este método, ya que los manipuladores se mueven constantemente a lo largo de regiones de su espacio de trabajo tan ampliamente separadas que no puede encontrarse una linealización válida para todas las regiones.

Otro método es desplazar el punto de operación con el manipulador a medida que se vaya moviendo, siempre linealizando sobre la posición deseada del manipulador. El resultado de este tipo de *linealización móvil* es un sistema lineal, pero variante en el tiempo. Aunque esta linealización casi estática del sistema original es útil en ciertas técnicas de análisis y diseño, no la utilizaremos en nuestro procedimiento de síntesis de leyes de control; trataremos directamente con las ecuaciones no lineales de movimiento y no recurriremos a las linealizaciones para derivar un controlador.

Si el resorte de la figura 9.6 en lugar de ser lineal tuviera un elemento no lineal, podríamos considerar el sistema como casi estático y, en cada instante, averiguar en dónde se ubican los polos del sistema. Podríamos descubrir que los polos se "mueven" alrededor del plano real – imaginario como una función de la posición del bloque. Por lo tanto, no podríamos seleccionar ganancias fijas que mantuvieran a los polos en una ubicación deseada (por ejemplo, en amortiguamiento crítico). Entonces podemos vernos tentados a considerar una ley de control más complicada, en la que las ganancias sean variantes en el tiempo (en realidad, varían como una función de la posición del bloque) de tal forma que el sistema siempre esté críticamente amortiguado. En esencia, para hacer esto se calcularía k_p de tal forma que la combinación del efecto no lineal del resorte se cancelara exactamente por un término no lineal en la ley de control, para que la rigidez general se mantuviera constante en todo momento. Dicho esquema de control podría llamarse ley de control de **linealización**, ya que utiliza un término de control no lineal para "cancelar" una no linealidad en el sistema controlado, de forma que el sistema de lazo cerrado en general sea lineal.

Ahora regresaremos a nuestra ley de control particionado y veremos que puede realizar esta función de linealización. En nuestro esquema de la ley de control particionado, la ley del servo permanece igual que siempre, pero la porción basada en el modelo ahora contendrá un modelo de la no linealidad. Por ende, la porción basada en el modelo del control realiza una función de linealización. Esto se entenderá mejor con un ejemplo.

EJEMPLO 10.1

Considere la característica del resorte no lineal mostrada en la figura 10.1. En lugar de la relación usual de resorte lineal, $f = kx$, este resorte se describe mediante $f = qx^3$. Si este resorte es parte del sistema físico que se muestra en la figura 9.6, construya una ley de control para mantener el sistema críticamente amortiguado con una rigidez de k_{CL}.

La ecuación de lazo abierto es

$$m\ddot{x} + b\dot{x} + qx^3 = f. \tag{10.1}$$

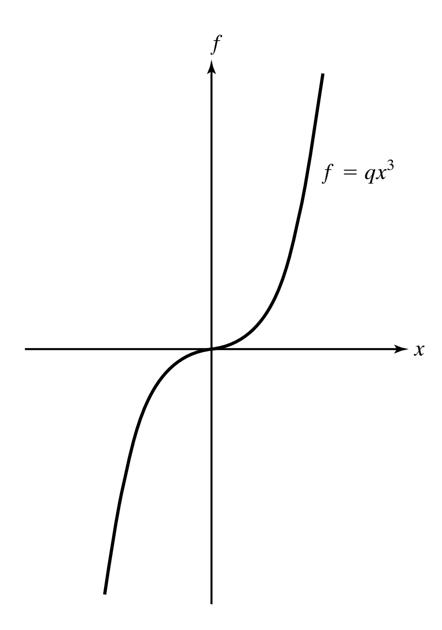


FIGURA 10.1: La característica de fuerza vs. distancia de un resorte no lineal.

La porción del control basada en el modelo es $f = \alpha f' + \beta$, en donde ahora utilizamos

$$\begin{aligned} \alpha &= m, \\ \beta &= b\dot{x} + qx^3; \end{aligned} \tag{10.2}$$

la porción de servo es, como siempre,

$$f' = \ddot{x}_d + k_v\dot{e} + k_p e, \tag{10.3}$$

en donde los valores de las ganancias se calculan a partir de cierta especificación del rendimiento deseado. La figura 10.2 muestra un diagrama de bloques de este sistema de control. El sistema resultante de lazo cerrado mantiene los polos en ubicaciones fijas.

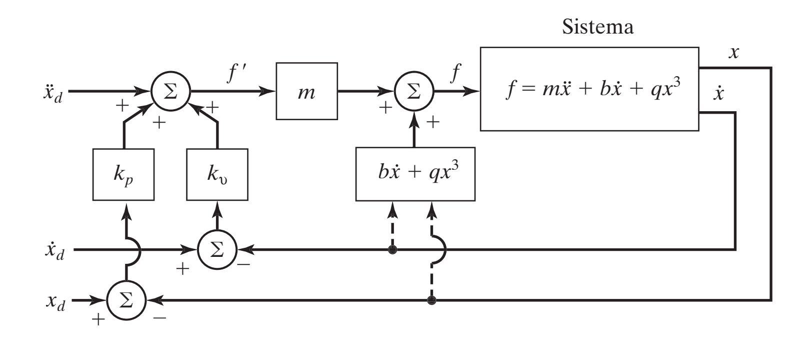


FIGURA 10.2: Un sistema de control no lineal para un sistema con un resorte no lineal.

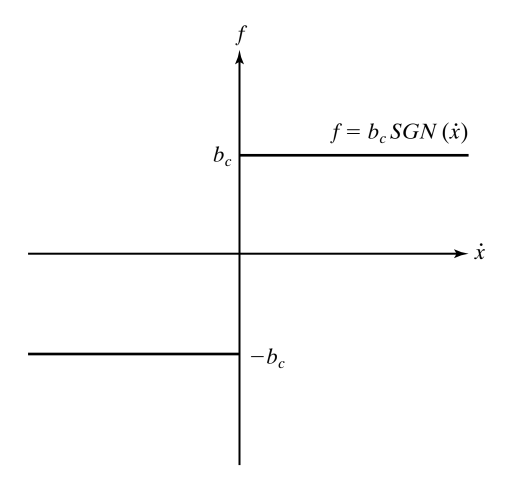


FIGURA 10.3: La característica de fuerza vs. velocidad de la fricción de Coulomb.

EJEMPLO 10.2

Considere la característica de la fricción no lineal que se muestra en la figura 10.3. Mientras que la fricción lineal está descrita por $f = b\dot{x}$, esta **fricción de Coulomb** se describe mediante $f = b_c sgn(\dot{x})$. Para la mayoría de los manipuladores actuales, la fricción de la articulación en su cojinete (sea giratorio o lineal) se modela con más precisión utilizando esta característica no lineal que con el modelo lineal más simple. Si este tipo de fricción está presente en el sistema de la figura 9.6, diseñe un sistema de control que utilice una porción basada en el modelo no lineal para amortiguar críticamente al sistema en todo momento.

La ecuación de lazo abierto es

$$m\ddot{x} + b_c sgn(\dot{x}) + kx = f. \tag{10.4}$$

La ley de control particionado es $f = \alpha f' + \beta$, en donde

$$\begin{aligned} \alpha &= m, \\ \beta &= b_c sgn(\dot{x}) + kx, \\ f' &= \ddot{x}_d + k_v \dot{e} + k_p e, \end{aligned} \tag{10.5}$$

en donde los valores de las ganancias se calculan a partir de una especificación de rendimiento deseado.

EJEMPLO 10.3

Considere el manipulador de un solo vínculo que se muestra en la figura 10.4; el cual tiene una articulación giratoria. Se considera que la masa está ubicada en un punto del extremo distal del vínculo, por lo que el momento de inercia es ml^2. Hay fricción de viscosidad y de Coulomb actuando en la articulación, y existe una carga debido a la gravedad.

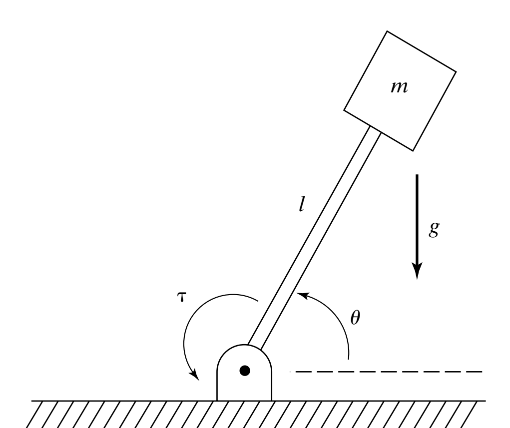


FIGURA 10.4: Un péndulo invertido, o un manipulador de un solo vínculo.

El modelo del manipulador es

$$\tau = ml^2\ddot{\theta} + \upsilon\dot{\theta} + csgn(\dot{\theta}) + mlg\cos(\theta). \tag{10.6}$$

Como siempre, el sistema de control tiene dos partes: alinealización de la porción de basada en el modelo y la porción de la ley del servo.

La porción del control basada en el modelo es $f = \alpha f' + \beta$, en donde

$$\alpha = ml^2,$$
$$\beta = \upsilon\dot{\theta} + csgn(\dot{\theta}) + mlg\cos(\theta); \tag{10.7}$$

la porción del servo es, como siempre,

$$f' = \ddot{\theta}_d + k_\upsilon \dot{e} + k_p e, \tag{10.8}$$

en donde los valores de las ganancias se calculan a partir de cierta especificación de rendimiento deseada.

Hemos visto que, en ciertos casos simples, no es difícil diseñar un controlador no lineal. El método general utilizado en los ejemplos simples antes mencionados es el mismo que utilizaremos para el problema de control de manipuladores:

1. Calcular una ley de control no lineal basada en el modelo que "cancele" las no linealidades del sistema a controlar.
2. Reducir el sistema a uno lineal que pueda controlarse con la ley de servo lineal simple desarrollada para la masa unitaria.

En cierto sentido, la ley de control de linealización implementa un *modelo inverso* del sistema que se está controlando. Las no linealidades del sistema cancelan las del modelo inverso; lo cual, en conjunto con la ley de servo, produce un sistema lineal de lazo cerrado. Obviamente, para hacer esta cancelación debemos conocer los parámetros y la estructura del sistema no lineal. Esto a menudo representa un problema en la aplicación práctica de este método.

10.3 SISTEMAS DE CONTROL MULTIENTRADAS, MULTISALIDAS

A diferencia de los ejemplos simples que hemos visto hasta ahora, el problema de controlar un manipulador es un problema de multientradas, multisalidas (MIMO). Es decir, tenemos un *vector* de posiciones, velocidades y aceleraciones de la articulación deseadas, y la ley de control debe calcular un *vector* de señales de articulación-actuador. Nuestro esquema básico en el que se particiona la ley de control en una porción basada en el modelo y una porción del servo aún puede aplicarse, pero ahora aparece en una forma matricial-vectorial. La ley de control toma la forma

$$F = \alpha F' + \beta, \tag{10.9}$$

en donde, para un sistema de n grados de libertad, F, F' y β son vectores de $n \times 1$ y α es una matriz de $n \times n$. Observe que la matriz α no es necesariamente diagonal, sino que se elige para **desacoplar** las n ecuaciones de movimiento. Si α y β se seleccionan correctamente, entonces, desde la entrada F', el sistema parece como si fuera n masas unitarias independientes. Por esta razón, en el caso multidimensional la porción basada en el modelo de la ley de control se llama ley de control de **linealización y desacoplamiento**. La ley de servo para un sistema multidimensional se convierte en

$$F' = \ddot{X}_d + K_v \dot{E} + K_p E, \tag{10.10}$$

en donde K_v y K_p son ahora matrices de $n \times n$, que se seleccionan generalmente para ser diagonales con ganancia constante en la diagonal. E y $\dot{E}$ son los vectores de $n \times 1$ de los errores en posición y velocidad, respectivamente.

10.4 EL PROBLEMA DEL CONTROL PARA LOS MANIPULADORES

En el caso del control de manipuladores, en el capítulo 6 desarrollamos un modelo y las correspondientes ecuaciones de movimiento. Como vimos, esas ecuaciones son bastante complicadas. La dinámica de cuerpo rígido tiene la forma

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta), \tag{10.11}$$

en donde $M(\Theta)$ es la matriz de inercia de $n \times n$ del manipulador, $V(\Theta, \dot{\Theta})$ es un vector de $n \times 1$ de términos centrífugos y de Coriolis, y $G(\Theta)$ es un vector de $n \times 1$ de términos de gravedad. Cada elemento de $M(\Theta)$ y de $G(\Theta)$ es una función complicada que depende de Θ, la posición de todas las articulaciones del manipulador. Cada elemento de $V(\Theta, \dot{\Theta})$ es una función complicada tanto de Θ como de $\dot{\Theta}$.

Además, podríamos incorporar un modelo de fricción (u otros efectos de cuerpos no rígidos). Suponiendo que nuestro modelo de fricción es una función de las posiciones y las velocidades de las articulaciones, agregamos el término $F(\Theta,\dot{\Theta})$ a la ecuación (10.11) para producir el modelo

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta) + F(\Theta, \dot{\Theta}). \tag{10.12}$$

El problema de controlar un sistema complicado, como el de la ecuación (10.12), puede manejarse mediante el esquema del controlador particionado que hemos presentado en este capítulo. En este caso, tenemos que

$$\tau = \alpha \tau' + \beta, \tag{10.13}$$

en donde τ es el vector de $n \times 1$ de momentos de torsión de las articulaciones. Elegimos lo siguiente:

$$\begin{aligned} \alpha &= M(\Theta), \\ \beta &= V(\Theta, \dot{\Theta}) + G(\Theta) + F(\Theta, \dot{\Theta}), \end{aligned} \tag{10.14}$$

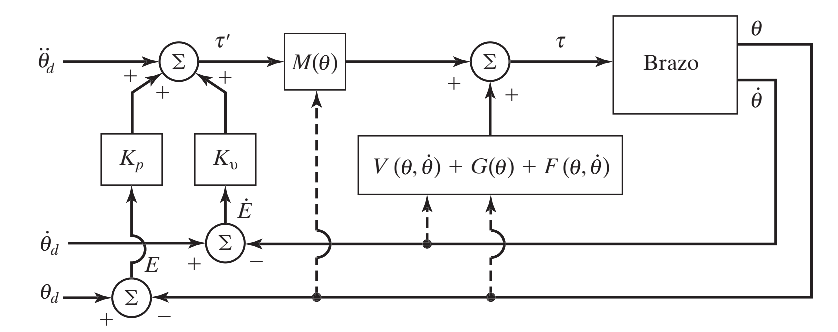


FIGURA 10.5: Un sistema de control de manipulador basado en el modelo.

con la ley de servo

$$\tau' = \ddot{\Theta}_d + K_v \dot{E} + K_p E, \tag{10.15}$$

en donde

$$E = \Theta_d - \Theta. \tag{10.16}$$

El sistema de control resultante se muestra en la figura 10.5.

Utilizando las ecuaciones (10.12) a (10.15), es muy fácil demostrar que el sistema de lazo cerrado está caracterizado por la ecuación de error

$$\ddot{E} + K_v \dot{E} + K_p E = 0. \tag{10.17}$$

Observe que esta ecuación vectorial está desacoplada: las matrices K_v y K_p son diagonales, de manera que la ecuación (10.17) podría escribirse también con base en cada articulación de la siguiente forma:

$$\ddot{e}_i + k_{vi} \dot{e} + k_{pi} e = 0. \tag{10.18}$$

El rendimiento ideal, representado por la ecuación (10.17), no puede obtenerse en la práctica por diversas razones, entre las cuales las más importantes son:

1. La naturaleza discreta de una implementación en computadora digital, a diferencia de la ley de control ideal de tiempo continuo implicada por las ecuaciones (10.14) y (10.15).
2. Imprecisión en el modelo del manipulador [necesario para evaluar la ecuación (10.14)].

En la siguiente sección trataremos (al menos parcialmente) estos dos temas.

10.5 CONSIDERACIONES PRÁCTICAS

Al desarrollar el control de linealización y desacoplamiento en las últimas secciones, hemos hecho implícitamente unas cuantas suposiciones que raras veces son ciertas en la práctica.

Tiempo requerido para calcular el modelo

En todas nuestras consideraciones de la estrategia de la ley de control particionado hemos supuesto implícitamente que todo el sistema estaba funcionando en tiempo continuo y que los cálculos en la ley de control requerían de un tiempo igual a cero. Dada cualquier cantidad de cálculos, con una computadora lo suficientemente grande, podemos realizar los cálculos con la suficiente rapidez como para que ésta sea una aproxi-

mación razonable; sin embargo, el costo de la computadora podría hacer que el esquema fuera económicamente inviable. En el caso del control del manipulador, toda la ecuación dinámica del manipulador (10.14) debe calcularse en la ley de control. Estos cálculos son considerablemente complejos; por consecuencia y como se dijo en el capítulo 6, hay un gran interés en desarrollar esquemas computacionales veloces para realizar los cálculos de una manera eficiente. A medida que el poder de cómputo se vuelva cada vez más asequible, las leyes de control que requieran de mucho poder de cómputo se volverán más prácticas. Se han reportado varias implementaciones experimentales de las leyes de control basadas en modelos no lineales [5–9] y las implementaciones parciales han empezado a aparecer en controladores industriales.

Como vimos en el capítulo 9, casi todos los sistemas de control de manipuladores operan ahora en circuitos digitales y se ejecutan a cierta **velocidad de muestreo**. Esto significa que los sensores de posición (y posiblemente otros) se leen en puntos discretos en el tiempo. De los valores leídos se calcula un comando que se envía al actuador. Por ende, las operaciones de lectura de los sensores y del envío de comandos al actuador no se hacen continuamente, sino a una velocidad de muestreo finita. Para analizar el efecto del retraso debido a los cálculos y la velocidad de muestreo finita, debemos usar herramientas del campo del **control de tiempo discreto**, en donde las ecuaciones diferenciales se convierten en ecuaciones de diferencia, y se ha desarrollado un conjunto completo de herramientas para responder a las preguntas sobre estabilidad y colocación de los polos para estos sistemas. La teoría de control de tiempo discreto está más allá del alcance de este libro, aunque para los investigadores que trabajan en el área del control de manipuladores muchos de estos conceptos son esenciales (vea la referencia [10]).

Aunque son importantes, las ideas y los métodos de la teoría de control de tiempo discreto son algo difíciles de aplicar en el caso de los sistemas no lineales. Aunque hemos logrado escribir una complicada ecuación diferencial de movimiento para la ecuación dinámica de un manipulador, una ecuación equivalente en tiempo discreto sería imposible de obtener, ya que para un manipulador general la única manera de resolver el movimiento del manipulador para un conjunto dado de condiciones iniciales, una entrada y un intervalo finito, es por medio de la integración numérica (como vimos en el capítulo 6). Es posible crear modelos de tiempo discreto si estamos dispuestos a utilizar soluciones en series para las ecuaciones diferenciales, o si hacemos aproximaciones. No obstante, si necesitamos hacer aproximaciones para desarrollar un modelo discreto, no está claro si tenemos un mejor modelo que el que tendríamos si utilizamos solamente el modelo continuo y hacemos la aproximación de tiempo continuo. Basta decir que es difícil analizar el problema de control de manipuladores en tiempo discreto, y por lo general se recurre a la simulación para poder juzgar el efecto que cierta velocidad de muestreo tendrá sobre el rendimiento.

En general supondremos que los cálculos pueden realizarse lo suficientemente rápido y con la suficiente frecuencia como para que sea válida la aproximación en tiempo continuo.

Control no lineal de alimentación anticipada

El uso del **control de alimentación anticipada** se ha propuesto como un método para usar un modelo dinámico no lineal en una ley de control sin necesidad de que se realicen cálculos complejos que consuman mucho tiempo a las velocidades de servo [11]. En la figura 10.5 la porción de control basada en el modelo de la ley de control está en el "ciclo de servo", en donde las señales "fluyen" a través de la caja negra con cada pulso

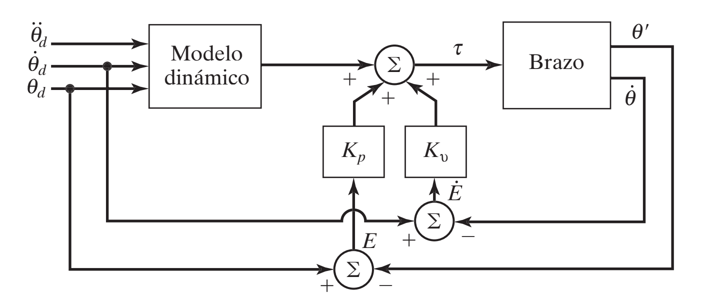


FIGURA 10.6: Esquema de control con la porción basada en el modelo "fuera" del ciclo de servo.

del reloj del servo. Si deseamos seleccionar una velocidad de muestreo de 200 Hz, entonces el modelo dinámico del manipulador debe calcularse a esa velocidad. Otro posible sistema de control se muestra en la figura 10.6. Aquí el control basado en el modelo está "fuera" del ciclo de servo. Por ende, es posible tener un ciclo de servo interno más rápido, que consista simplemente de multiplicar los errores por las ganancias, agregando los momentos de torsión basados en el modelo a una velocidad menor.

Desafortunadamente, el esquema de alimentación anticipada de la figura 10.6 no proporciona un desacoplamiento completo. Si escribimos las ecuaciones del sistema,[1] encontraremos que la ecuación de error de este sistema es

$$\ddot{E} + M^{-1}(\Theta)K_\upsilon \dot{E} + M^{-1}(\Theta)K_p E = 0. \tag{10.19}$$

Evidentemente, a medida que cambie la configuración del brazo cambiará la ganancia efectiva de lazo cerrado y los polos casi estáticos se desplazarán en el plano real-imaginario. No obstante, la ecuación (10.19) podría utilizarse como punto de inicio para diseñar un **controlador robusto** que encuentre un buen conjunto de ganancias constantes de tal forma que, a pesar del "movimiento" de los polos, se garantice que permanezcan en ubicaciones razonablemente favorables. Alternativamente podríamos considerar esquemas en los que se calculen previamente las ganancias variables que cambien con la configuración del robot, de manera que los polos casi estáticos del sistema permanezcan en posiciones fijas.

Observe que en el sistema de la figura 10.6, el modelo dinámico se calcula solamente como una función de la ruta deseada, por lo que cuando se conoce de antemano la ruta, los valores pueden calcularse "fuera de línea" antes de que empiece el movimiento. En tiempo de ejecución los historiales de momentos de torsión previamente calculados se leerían de la memoria. De igual forma, si se calculan ganancias variantes en el tiempo, también podrían calcularse de antemano y almacenarse. Por ende, un esquema así podría ser considerablemente económico en tiempo de ejecución, hablando en términos computacionales, y por lo tanto podría lograr una velocidad de servo alta.

Implementación de momento de torsión calculado con doble velocidad

La figura 10.7 muestra el diagrama de bloques de una posible implementación práctica del sistema de control de posición de linealización y desacoplamiento. El modelo dinámico se expresa en su forma de *espacio de configuración*, de tal forma que los parámetros dinámicos del manipulador aparezcan como funciones de la posición del manipulador

[1] Hemos usado las siguiente suposiciones para simplificar: $M(\Theta_d) \cong M(\Theta)$, $V(\Theta_d, \dot{\Theta}_d) \cong V(\Theta, \dot{\Theta})$, $G(\Theta_d) \cong G(\Theta)$, y $F(\Theta_d, \dot{\Theta}_d) \cong F(\Theta, \dot{\Theta})$.

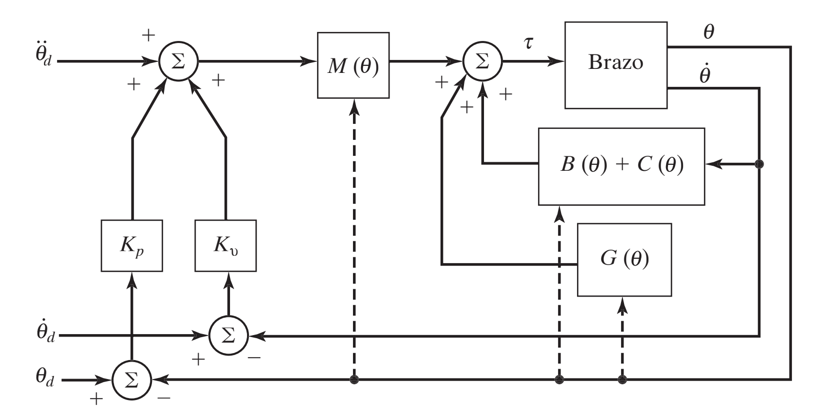


FIGURA 10.7: Una implementación del sistema de control de manipulador basado en el modelo.

solamente. Estas funciones podrían entonces calcularse mediante un proceso en *segundo plano* o mediante una segunda computadora de control [8], o podrían buscarse en una tabla previamente calculada [12]. En esta arquitectura, los parámetros dinámicos pueden actualizarse a una velocidad menor que la del servo de lazo cerrado. Por ejemplo, el cálculo en segundo plano podría llevarse a cabo a 60Hz, mientras que el servo de lazo cerrado estuviera funcionando a 250 Hz.

Desconocimiento de los parámetros

La segunda dificultad potencial que podemos encontrar al emplear el algoritmo de control de momento de torsión calculado es que, por lo general, el modelo dinámico del manipulador no se conoce con precisión. Esto es especialmente cierto en algunos componentes de la dinámica, como los efectos de la fricción. De hecho, es muy difícil conocer la estructura del modelo de fricción, por no hablar de los valores de los parámetros [13]. Por último, si el manipulador tiene cierta porción de su dinámica que no puede repetirse (por ejemplo, porque cambie a medida que el robot se desgasta), es difícil tener buenos valores para los parámetros del modelo en todo momento.

Por naturaleza, la mayoría de los robots estarán recogiendo varias piezas y herramientas. Cuando un robot sujeta una herramienta, la inercia y el peso de la misma cambian la dinámica del manipulador. En una situación industrial, las propiedades de la masa de las herramientas podrían conocerse; en este caso, pueden incluirse en la porción modelada de la ley de control. Al sujetar una herramienta, la matriz de inercia, la masa total y el centro de la masa del último vínculo del manipulador pueden actualizarse con nuevos valores que representan el efecto combinado del último vínculo más la herramienta. No obstante, en muchas aplicaciones las propiedades de las masas de los objetos que recoge el manipulador no se conocen, por lo que es difícil mantener un modelo dinámico preciso.

La situación no ideal más simple posible es aquella en la que aún suponemos un modelo perfecto implementado en tiempo continuo, pero con un ruido externo que actúa para perturbar el sistema. En la figura 10.8 indicamos un vector de momentos de torsión de perturbación que actúan en las articulaciones. Al escribir la ecuación de error del sistema, incluyendo estas perturbaciones conocidas, llegamos a

$$\ddot{E} + K_v \dot{E} + K_p E = M^{-1}(\Theta)\tau_d, \tag{10.20}$$

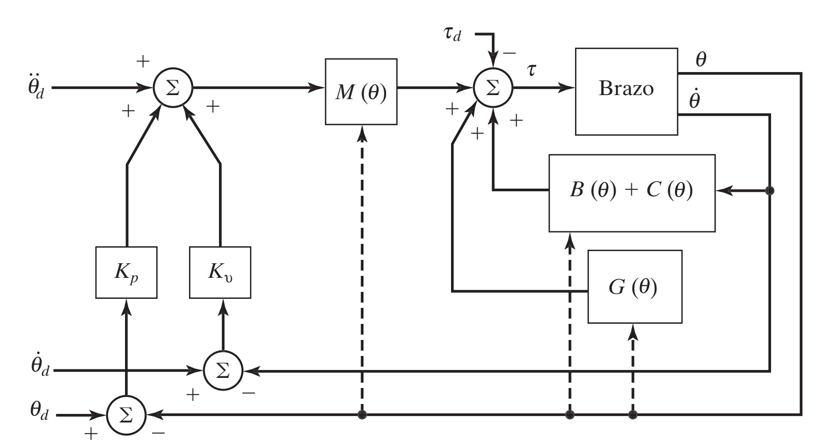


FIGURA 10.8: El controlador basado en el modelo con una perturbación externa en acción.

en donde τ_d es el vector de momentos de torsión de perturbación en las articulaciones. El lado izquierdo de la ecuación (10.20) está desacoplado, pero del lado derecho podemos ver que una perturbación en cualquier articulación específica introducirá errores en las demás articulaciones, ya que $M(\Theta)$ no es, en general, diagonal.

Con base en la ecuación (10.20) podrían realizarse algunos análisis simples; por ejemplo, es fácil calcular el error de servo de estado estable debido a una perturbación constante como

$$E = K_p^{-1} M^{-1}(\Theta)\tau_d. \tag{10.21}$$

Cuando nuestro modelo del comportamiento dinámico de los manipuladores no es perfecto, el análisis del sistema de lazo cerrado resultante se hace más difícil. Definiremos la siguiente notación: $\hat{M}(\Theta)$ es nuestro modelo de la matriz de inercia del manipulador, $M(\Theta)$. De igual forma, $\hat{V}(\Theta, \dot{\Theta})$, $\hat{G}(\Theta)$ y $\hat{F}(\Theta, \dot{\Theta})$ son nuestros modelos de los términos de velocidad, de gravedad y de fricción del mecanismo actual respectivamente. Un conocimiento perfecto del modelo significaría que

$$\begin{aligned} \hat{M}(\Theta) &= M(\Theta), \\ \hat{V}(\Theta, \dot{\Theta}) &= V(\Theta, \dot{\Theta}), \\ \hat{G}(\Theta) &= G(\Theta), \\ \hat{F}(\Theta, \dot{\Theta}) &= F(\Theta, \dot{\Theta}). \end{aligned} \tag{10.22}$$

Por lo tanto, y aunque la dinámica de manipuladores se da mediante

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta) + F(\Theta, \dot{\Theta}), \tag{10.23}$$

nuestra ley de control calcula

$$\begin{aligned} \tau &= \alpha\tau' + \beta, \\ \alpha &= \hat{M}(\Theta), \\ \beta &= \hat{V}(\Theta, \dot{\Theta}) + \hat{G}(\Theta) + \hat{F}(\Theta, \dot{\Theta}). \end{aligned} \tag{10.24}$$

Por lo tanto, el desacoplamiento y la linealización no se realizarán perfectamente cuando los parámetros no se conozcan con exactitud. Si escribimos la ecuación de lazo cerrado para el sistema, tenemos que

$$\ddot{E} + K_\upsilon \dot{E} + K_p E = \hat{M}^{-1}[(M - \hat{M})\ddot{\Theta} + (V - \hat{V}) + (G - \hat{G}) + (F - \hat{F})], \tag{10.25}$$

en donde los argumentos de las funciones dinámicas no se muestran por cuestión de brevedad. Observe que si el modelo fuera exacto, de manera que la ecuación (10.22) fuera cierta, entonces el lado derecho de la ecuación (10.25) sería cero y los errores desaparecerían. Cuando los parámetros no se conocen con exactitud, la diferencia entre los parámetros actuales y los modelados producirá la excitación de los errores del servo (lo que posiblemente produzca incluso un sistema inestable [21]) de acuerdo con la ecuación (10.25), que es algo complicada.

Hablaremos sobre el análisis de estabilidad de un sistema de lazo cerrado no lineal en la sección 10.7.

10.6 SISTEMAS ACTUALES DE CONTROL DE ROBOTS INDUSTRIALES

Debido a los problemas para tener un buen conocimiento de los parámetros, no está claro si tiene sentido tomarse la molestia de calcular una ley complicada de control basada en el modelo para el control de manipuladores. El costo del poder de cómputo necesario para calcular el modelo del manipulador a una velocidad suficiente podría no valer la pena, especialmente cuando la falta de conocimiento de los parámetros pudiera anular los beneficios de dicho método. Los fabricantes de robots industriales han decidido, probablemente por razones económicas, que no vale la pena tratar de usar un modelo de manipulador completo en el controlador. Por ahora los manipuladores actuales se controlan mediante leyes de control muy simples que por lo general están completamente controladas por errores y se implementan en arquitecturas como las que estudiamos en la sección 9.10. En la figura 10.9 se muestra un robot industrial con un sistema servo de alto rendimiento.

Control PID de una articulación

La mayoría de los robots industriales actuales tienen un esquema de control que, en nuestra notación, estaría descrito por

$$\alpha = I, \qquad \beta = 0, \tag{10.26}$$

en donde I es la matriz identidad de $n \times n$. La porción del servo es

$$\tau' = \ddot{\Theta}_d + K_\upsilon \dot{E} + K_p E + K_i \int E dt, \tag{10.27}$$

en donde K_υ, K_p y K_i son matrices diagonales constantes. En muchos casos, $\ddot{\Theta}_d$ no está disponible y este término se hace simplemente cero. Esto es, la mayoría de los controladores de robots no utilizan un componente basado en el modelo *en ninguna parte* de su ley de control. Este tipo de esquema de control PID es simple, ya que cada articulación se controla como un sistema de control separado. A menudo se utiliza un microprocesador por cada articulación para implementar la ecuación (10.27), como se vio en la sección 9.10.



FIGURA 10.9: El Adept One, un robot de control directo fabricado por Adept Technology, Inc.

El rendimiento de un manipulador controlado de esta forma no es fácil de describir. No se está realizando un desacoplamiento, por lo que el movimiento de cada articulación afecta a las demás articulaciones. Estas interacciones producen errores, los cuales son suprimidos por la ley de control manejada por errores. Es imposible seleccionar ganancias fijas que amortigüen críticamente la respuesta a las perturbaciones para todas las configuraciones. Por lo tanto se seleccionan ganancias "promedio" que aproximan el amortiguamiento crítico en el centro del espacio de trabajo del robot. En varias configuraciones extremas del brazo, el sistema se vuelve ya sea subamortiguado o sobreamortiguado. Dependiendo de los detalles del diseño mecánico del robot, estos efectos podrían ser relativamente pequeños, y el control sería bueno. En dichos sistemas es importante mantener las ganancias lo más altas posible para que las inevitables perturbaciones se supriman rápidamente.

Adición de compensación de gravedad

Los términos de gravedad tienden a provocar errores de posicionamiento estático, por lo que algunos fabricantes de robots incluyen un modelo de gravedad $G(\theta)$ en la ley de control [esto es, $\beta = \hat{G}(\Theta)$ en nuestra notación]. La ley de control completa toma la forma

$$\tau' = \ddot{\Theta}_d + K_v \dot{E} + K_p E + K_i \int E dt + \hat{G}(\Theta). \tag{10.28}$$

Dicha ley de control es tal vez el ejemplo más simple de un controlador basado en el modelo. Como la ecuación (10.28) ya no puede implementarse estrictamente de articulación en articulación, la arquitectura del controlador debe permitir la comunicación entre los controladores de las articulaciones o debe hacer uso de un procesador central, en vez de utilizar un procesador para cada articulación.

Varias aproximaciones del control de desacoplamiento

Existen varias formas de simplificar las ecuaciones dinámicas de un manipulador específico [3, 14]. Después de la simplificación puede derivarse una ley de desacoplamiento y linealización aproximada. Una simplificación común podría ser descartar los componentes del momento de torsión debidos a los términos de velocidad; esto es, modelar solamente los términos de inercia y de gravedad. A menudo los modelos de fricción no se incluyen en el controlador, ya que la fricción es muy difícil de modelar correctamente. Algunas veces la matriz de inercia se simplifica para tomar en cuenta el acoplamiento mayor entre ejes, pero no para los efectos menores de acoplamiento cruzado. Por ejemplo, la referencia [14] presenta una versión simplificada de la matriz de masas del PUMA 560 que requiere sólo un 10% aproximadamente de los cálculos necesarios para calcular la matriz de masas completa, y aún así tiene una precisión dentro de un rango del 1%.

10.7 ANÁLISIS DE ESTABILIDAD DE LYAPUNOV

En el capítulo 9 examinamos los sistemas de control lineal en forma analítica para evaluar la estabilidad y también el rendimiento de la respuesta dinámica en términos de amortiguamiento y ancho de banda de lazo cerrado. Los mismos análisis son válidos para un sistema no lineal que haya sido desacoplado y linealizado por medio de un controlador no lineal perfecto basado en el modelo, ya que el sistema resultante general es otra vez lineal. No obstante, cuando el controlador no realiza el desacoplamiento y la linealización o cuando son incompletos o imprecisos, el sistema de lazo cerrado general sigue siendo no lineal. Para los sistemas no lineales es mucho más difícil realizar el análisis de estabilidad y rendimiento. En esta sección presentaremos un método de análisis de estabilidad que se aplica tanto a sistemas lineales como a sistemas no lineales.

Considere el sistema simple de masa-resorte con fricción que se presentó originalmente en el capítulo 9, cuya ecuación de movimiento es

$$m\ddot{x} + b\dot{x} + kx = 0. \tag{10.29}$$

La energía total del sistema se da mediante

$$\upsilon = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}kx^2, \tag{10.30}$$

en donde el primer término proporciona la energía cinética de la masa, y el segundo proporciona la energía potencial almacenada en el resorte. Observe que el valor υ de energía del sistema siempre es no negativo (es decir, es positivo o cero). Vamos a encontrar la *tasa* de cambio de la energía total al diferenciar la ecuación (10.30) respecto al tiempo, para obtener

$$\dot{\upsilon} = m\dot{x}\ddot{x} + kx\dot{x}. \tag{10.31}$$

Sustituyendo la ecuación (10.29) por $m\ddot{x}$ en la ecuación (10.31) obtenemos

$$\dot{\upsilon} = -b\dot{x}^2, \tag{10.32}$$

la cual podemos ver que es siempre no positiva (porque $b > 0$). Por lo tanto, la energía siempre sale del sistema, a menos que $\dot{x} = 0$. Esto implica que, sin importar la perturbación inicial, el sistema perderá energía hasta quedar inactivo. Investigando las posibles posiciones de reposo mediante un análisis de estado estable de la ecuación (10.29) obtenemos

$$kx = 0, \tag{10.33}$$

o

$$x = 0. \tag{10.34}$$

En consecuencia, por medio de un análisis de energía hemos demostrado que el sistema de la ecuación (10.29) con cualquier condición inicial (es decir, cualquier energía inicial) con el tiempo quedará inactivo en el punto de equilibrio. Esta prueba de estabilidad por medio de un análisis de energía es un simple ejemplo de una técnica más general conocida como **análisis de estabilidad de Lyapunov**, o **método segundo** (o **directo**) **de Lyapunov**, nombrado así en honor de un matemático ruso del siglo XIX [15].

Una característica interesante de este método de análisis de estabilidad es que podemos hacer conclusiones sobre la estabilidad sin despejar la solución de la ecuación diferencial que gobierna al sistema. No obstante, mientras que el método de Lyapunov es útil para examinar la *estabilidad*, generalmente no proporciona ninguna información sobre la respuesta transitoria o el *rendimiento* del sistema. Observe que nuestro análisis de energía no produjo información acerca de si el sistema estaba sobreamortiguado o subamortiguado, o sobre cuánto tiempo le llevaría suprimir una perturbación. Es importante distinguir entre estabilidad y rendimiento: un sistema estable podría no obstante exhibir un rendimiento de control insatisfactorio para su uso planeado.

El método de Lyapunov es algo más general que nuestro ejemplo. Es una de las pocas técnicas que pueden aplicarse directamente a sistemas no lineales para investigar su estabilidad. Como medio para tener rápidamente una idea del método de Lyapunov (en suficiente detalle para nuestras necesidades), analizaremos una muy breve introducción a la teoría y luego pasaremos directamente a varios ejemplos. Podrá encontrar un tratamiento más completo de la teoría de Lyapunov en las referencias [16] y [17].

El método de Lyapunov está enfocado en determinar la estabilidad de una ecuación diferencial

$$\dot{X} = f(X), \tag{10.35}$$

en donde X es $m \times 1$ y $f(\cdot)$ podría ser no lineal. Observe que las ecuaciones diferenciales de mayor orden pueden escribirse siempre como un conjunto de ecuaciones de primer orden en la forma de la ecuación (10.35). Para probar que un sistema es estable mediante el método de Lyapunov, es necesario proponer una función de energía generalizada $\upsilon(X)$, que tenga las siguientes propiedades:

1. Que $\upsilon(X)$ tenga primeras derivadas parciales continuas, y que $\upsilon(X) > 0$ para todas las X excepto cuando $\upsilon(0) = 0$.
2. Que $\dot{\upsilon}(X) \leq 0$. Aquí $\dot{\upsilon}(X)$ representa el cambio en $\upsilon(X)$ a lo largo de todas las trayectorias del sistema.

Estas propiedades podrían ser válidas solamente en cierta región; o podrían ser globales, con sus correspondientes resultados de estabilidad más débiles o más fuertes. La idea intuitiva es que se demuestre que una función de estado positiva definida “tipo energía” siempre disminuye o permanece constante; por ende, el sistema es estable en el sentido en que el tamaño del vector de estado está acotado.

Cuando $\dot{\upsilon}(X)$ es estrictamente menor que cero, la convergencia asintótica del estado al vector cero puede concluirse. El trabajo original de Lyapunov se extendió de

una manera importante gracias a LaSalle y Lefschetz [4], quienes demostraron que en ciertas situaciones, aun cuando $\dot{\upsilon}(X) \leq 0$ (observe la igualdad incluida), puede demostrarse la estabilidad asintótica. Para nuestros fines podemos tratar con el caso en que $\dot{\upsilon}(X) = 0$ realizando un análisis de estado estable para saber si la estabilidad es asintótica o si el sistema que se está estudiando puede "atorarse" en algún lugar además de $\upsilon(X) = 0$.

Se dice que un sistema como el descrito por la ecuación (10.35) es **autónomo**, ya que la función $f(\cdot)$ no es explícita del tiempo. El método de Lyapunov se extiende también a los sistemas **no autónomos**, en los que el tiempo es un argumento de la función no lineal. Vea las referencias [4, 17] para más detalles.

EJEMPLO 10.4

Considere el sistema lineal

$$\dot{X} = -AX, \tag{10.36}$$

en donde A es de $m \times m$ y es positiva definida. Proponga la **función candidata de Lyapunov**

$$\upsilon(X) = \frac{1}{2}X^T X, \tag{10.37}$$

que es continua y no negativa en todas partes. La diferenciación produce

$$\begin{aligned} \dot{\upsilon}(X) &= X^T \dot{X} \\ &= X^T(-AX) \\ &= -X^T AX, \end{aligned} \tag{10.38}$$

que en todas partes es no positiva, ya que A es una matriz positiva definida. Por ende, la ecuación (10.37) es evidentemente una función de Lyapunov para el sistema de la ecuación (10.36). El sistema es asintóticamente estable, ya que $\dot{\upsilon}(X)$ puede ser cero sólo cuando $X = 0$; en cualquier otra parte, X debe disminuir.

EJEMPLO 10.5

Considere un sistema mecánico de resorte–amortiguador en el que tanto el resorte como el amortiguador son no lineales:

$$\ddot{x} + b(\dot{x}) + k(x) = 0. \tag{10.39}$$

Las funciones $b(\cdot)$ y $k(\cdot)$ son funciones continuas del primer y tercer cuadrantes, de tal forma que

$$\begin{aligned} &\dot{x}b(\dot{x}) > 0 \;\; f \; o \neq 0, \\ &xk(x) > 0 \;\; f \; o \; x \neq 0. \end{aligned} \tag{10.40}$$

Una vez propuesta la función de Lyapunov

$$\upsilon(x, \dot{x}) = \frac{1}{2}\dot{x}^2 + \int_0^x k(\lambda)d\lambda, \tag{10.41}$$

nos lleva a

$$\begin{aligned}\dot{\upsilon}(x,\dot{x}) &= \dot{x}\ddot{x} + k(x)\dot{x},\\ &= -\dot{x}b(\dot{x}) - k(x)\dot{x} + k(x)\dot{x},\\ &= -\dot{x}b(\dot{x}).\end{aligned} \tag{10.42}$$

Aquí, $\dot{\upsilon}(\cdot)$ es no positiva pero sólo está semidefinida, ya que no es una función de x sino sólo de $\dot{x}$. Para poder concluir una estabilidad asintótica, tenemos que asegurarnos que no es posible que el sistema se "atore" con una x distinta de cero. Para estudiar todas las trayectorias para las cuales $\dot{x} = 0$, debemos considerar que

$$\ddot{x} = -k(x), \tag{10.43}$$

en donde $x = 0$ es la única solución. Por ende, el sistema quedará inactivo sólo si $x = \dot{x} = \ddot{x} = 0$.

EJEMPLO 10.6

Considere un manipulador con una dinámica dada por

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta,\dot{\Theta}) + G(\Theta) \tag{10.44}$$

y controlado con la ley de control

$$\tau = K_p E - K_d\dot{\Theta} + G(\Theta), \tag{10.45}$$

en donde K_p y K_d son matrices de ganancia diagonales. Observe que este controlador no obliga al manipulador a seguir una trayectoria, sino que mueve el manipulador hasta un punto de destino sobre una ruta especificada por la dinámica del manipulador, y después regula la posición ahí. El sistema de lazo cerrado resultante, que se obtiene al igualar las ecuaciones (10.44) y (10.45), es

$$M(\Theta)\ddot{\Theta} + V(\Theta,\dot{\Theta}) + K_d\dot{\Theta} + K_p\Theta = K_p\Theta_d; \tag{10.46}$$

y se puede probar que es estable asintóticamente de manera global mediante el método de Lyapunov [18, 19].

Considere la siguiente función candidata de Lyapunov:

$$\upsilon = \frac{1}{2}\dot{\Theta}^T M(\Theta)\dot{\Theta} + \frac{1}{2}E^T K_p E. \tag{10.47}$$

La función (10.47) es siempre positiva o cero, ya que la matriz de masas del manipulador, representada por $M(\Theta)$, y la matriz de ganancia de posición, K_p, son matrices positivas definidas. Al diferenciar la ecuación (10.46) se produce la siguiente ecuación:

$$\begin{aligned}\dot{\upsilon} &= \frac{1}{2}\dot{\Theta}^T\dot{M}(\Theta)\dot{\theta} + \dot{\theta}^T M(\theta)\ddot{\Theta} - E^T K_p\dot{\Theta}\\ &= \frac{1}{2}\dot{\Theta}^T\dot{M}(\Theta)\dot{\Theta} - \dot{\Theta}^T K_d\dot{\Theta} - \dot{\Theta}^T V(\Theta,\dot{\Theta})\\ &= -\dot{\Theta}^T K_d\dot{\Theta},\end{aligned} \tag{10.48}$$

la cual será no positiva siempre y cuando K_d sea positiva definida. Para realizar el último paso en la ecuación (10.48) hemos utilizado la interesante identidad

$$\frac{1}{2}\dot{\Theta}^T \dot{M}(\Theta)\dot{\Theta} = \dot{\Theta}^T V(\Theta, \dot{\Theta}), \tag{10.49}$$

que puede demostrarse mediante la investigación de la estructura de las ecuaciones de movimiento de Lagrange [18–20]. (Vea también el ejercicio 6.17).

A continuación investigaremos si el sistema puede "atorarse" con un error distinto de cero. Como $\dot{\upsilon}$ puede ser cero sólo a lo largo de trayectorias en las que $\dot{\Theta} = 0$ y $\ddot{\Theta} = 0$, podemos ver de la ecuación (10.46) que, en este caso,

$$K_p E = 0, \tag{10.50}$$

y como K_p es no singular, tenemos que

$$E = 0. \tag{10.51}$$

Por lo tanto, la ley de control (10.45) aplicada al sistema (10.44) logra una estabilidad asintótica global.

Esta prueba es importante, ya que explica, hasta cierto punto, por qué funcionan los robots industriales de la actualidad, la mayoría de los cuales usa un servo simple controlado por errores, ocasionalmente con modelos de gravedad, y que por lo tanto son bastante similares a la ecuación (10.45).

Vea los ejercicios 10.11 a 10.16 para más ejemplos de leyes de control de manipuladores no lineales que puedan probarse estables por el método de Lyapunov. Recientemente, la teoría de Lyapunov ha aparecido cada vez más en las publicaciones de investigación de robótica [18–25].

10.8 SISTEMAS DE CONTROL DE BASE CARTESIANA

En esta sección presentaremos la noción de **control de base cartesiana**. Aunque estos métodos no se utilizan actualmente en los robots industriales, hay actividad en varias instituciones de investigación sobre dichos temas.

Comparación con los esquemas basados en articulaciones

En todos los esquemas de control para los manipuladores que hemos descrito hasta ahora, hemos supuesto que la trayectoria deseada estaba disponible en términos de historiales de tiempo de la posición, la velocidad y la aceleración de las articulaciones. Dado que estas entradas deseadas estaban disponibles, diseñamos esquemas de **control basado en articulaciones**; esto es, esquemas en los que evaluamos errores en la trayectoria al encontrar la diferencia entre las cantidades deseada y real, expresadas en el espacio de articulación. Muy a menudo deseamos que el efector final del manipulador siga líneas rectas u otras formas de ruta descritas en coordenadas cartesianas. Como vimos en el capítulo 7, es posible calcular los historiales de tiempo de la trayectoria de espacio de articulación que corresponda a las rutas de línea recta cartesianas. La figura 10.10 muestra este método para el control de trayectorias de manipuladores. Una característica básica del método es el proceso de **conversión de trayectorias**, el cual se utiliza para calcular las trayectorias de las articulaciones, lo cual va seguido por algún tipo de esquema de servo basado en la articulación, como el que hemos estado estudiando.

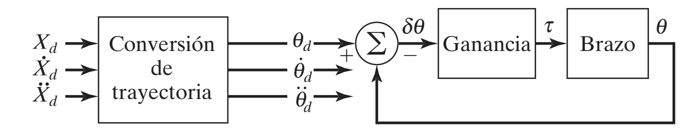


FIGURA 10.10: Un esquema de control basado en articulación, con entrada de ruta cartesiana.

El proceso de conversión de trayectorias es considerablemente difícil (en términos de costo computacional) si se va a realizar en forma analítica. Los cálculos que se requieren para ello son

$$\begin{aligned}\Theta_d &= CININV(\chi_d),\\ \dot{\Theta}_d &= J^{-1}(\Theta)\dot{\chi}_d, \qquad (10.52)\\ \ddot{\Theta}_d &= \dot{J}^{-1}(\Theta)\dot{\chi}_d + J^{-1}(\Theta)\ddot{\chi}_d.\end{aligned}$$

En la medida en que dicho cálculo se realice en los sistemas actuales, generalmente sólo se lleva a cabo la solución para Θ_d usando la cinemática inversa, y luego se calculan las velocidades y las aceleraciones de las articulaciones numéricamente mediante la primera y la segunda diferencias. No obstante, dicha diferenciación numérica tiende a amplificar el ruido e introduce un retraso, a menos que pueda realizarse con un filtro no causal.[2] Por lo tanto, estamos interesados en encontrar una manera menos costosa en términos computacionales de calcular la ecuación (10.52), o de sugerir un esquema de control en el que esta información no sea necesaria.

Un método alternativo se muestra en la figura 10.11. Aquí la posición detectada del manipulador se transforma inmediatamente por medio de las ecuaciones cinemáticas, en una descripción cartesiana de posición. Esta descripción se compara entonces con la posición cartesiana deseada para poder formar errores en espacio cartesiano. Los esquemas de control basados en la formación de errores en espacio cartesiano se conocen como esquemas de **control de base cartesiana**. Por simplicidad, en la figura 10.11 no se muestra la retroalimentación de la velocidad, pero estaría presente en cualquier implementación.

El proceso de conversión de trayectorias se sustituye por cierto tipo de conversión de coordenadas dentro del lazo del servo. Observe que los controladores de base cartesiana deben realizar muchos cálculos en el lazo; la cinemática y otras transformaciones se encuentran ahora "dentro del lazo". Esto puede representar una desventaja de los métodos de base cartesiana; el sistema resultante podría operar a una frecuencia

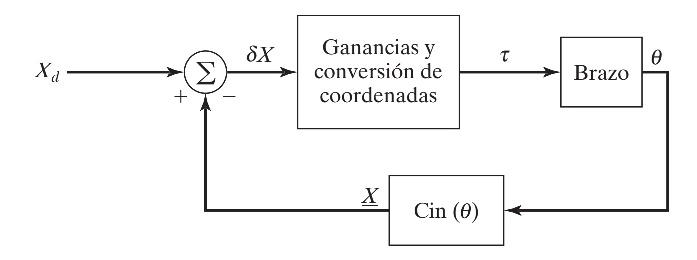


FIGURA 10.11: El concepto de un esquema de control de base cartesiana.

[2] La diferenciación numérica introduce un retraso a menos que pueda basarse en valores pasados, presentes y futuros. Cuando se planea previamente toda la ruta completa, puede hacerse este tipo de diferenciación numérica no causal.

de muestreo menor, en comparación con los sistemas basados en articulaciones (dada la misma capacidad de computadora), lo que degradaría en general la estabilidad y las capacidades del sistema para eliminar perturbaciones.

Esquemas intuitivos de control cartesiano

La figura 10.12 muestra un posible esquema de control que viene a la mente de manera algo intuitiva. Aquí la posición cartesiana se compara con la posición deseada para formar un error, representado por δX, en espacio cartesiano. Este error, que puede considerarse pequeño si el sistema de control está haciendo su trabajo, puede asignarse a un pequeño desplazamiento en espacio de articulación mediante el jacobiano inverso. Los errores resultantes en espacio de articulación, representados por $\delta\theta$, se multiplican entonces por las ganancias para calcular momentos de torsión que tiendan a reducir estos errores. Observe que la figura 10.12 muestra un controlador simplificado en el que, por claridad, no se ha mostrado la retroalimentación de velocidad, y que podría agregarse de una manera simple. Llamaremos a este esquema el **controlador de jacobiano inverso**.

La figura 10.13 muestra otro esquema que podría venir a la mente. Aquí el vector de error cartesiano se multiplica por una ganancia para calcular un vector de fuerza cartesiano. Esto puede pensarse como una fuerza cartesiana que, si se aplica al efector final del robot, empujaría al efector final en una dirección que tendería a reducir el error cartesiano. Este vector de fuerza cartesiana (que en realidad es un vector de fuerza-momento) se asigna entonces a través de la transpuesta del jacobiano para poder calcular los momentos de torsión de articulación equivalentes que tiendan a reducir los errores observados. A este esquema lo llamaremos **controlador de jacobiano transpuesto**.

Hemos llegado al controlador de jacobiano inverso y al controlador de jacobiano transpuesto de forma intuitiva. No podemos estar seguros de que dichos arreglos sean estables, mucho menos que tengan un buen rendimiento. Además, es curioso que los esquemas sean en extremo similares, excepto que uno contiene la inversa del jacobiano y el otro su transpuesta. Recuerde que la inversa no es igual a la transpuesta en general (sólo en el caso de un manipulador estrictamente cartesiano se aplica que $J^T = J^{-1}$). El rendimiento dinámico exacto de dichos sistemas (si se expresan en una ecuación de espacio de error de segundo orden, por ejemplo) es muy complicado; aunque resulta que

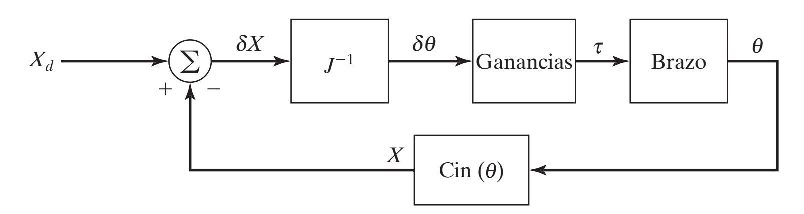


FIGURA 10.12: El esquema de control cartesiano de jacobiano inverso.

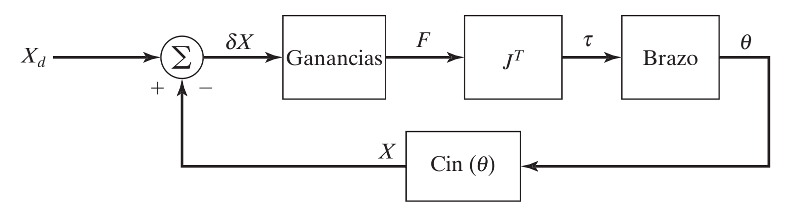


FIGURA 10.13: El esquema de control cartesiano de jacobiano transpuesto.

ambos esquemas funcionarán (es decir, pueden hacerse estables), pero no bien (es decir, el rendimiento no es bueno sobre todo el espacio de trabajo). Ambos pueden hacerse estables mediante la selección apropiada de ganancias, incluyendo cierta forma de retroalimentación de velocidad (la cual no se mostró en las figuras 10.12 y 10.13). Aunque ambos funcionan, ninguno es *correcto* en el sentido de que no podemos seleccionar ganancias fijas que produzcan polos de lazo cerrado fijos. La respuesta dinámica de dichos controladores variará con la configuración del brazo.

Esquema de desacoplamiento cartesiano

Para los controladores de base cartesiana tales como los basados en articulación, el buen rendimiento estaría caracterizado por una dinámica de error constante sobre todas las configuraciones del manipulador. Los errores se expresan en espacio cartesiano en esquemas de base cartesiana, lo que significa que sería conveniente diseñar un sistema que suprimiera los errores cartesianos de una manera críticamente amortiguada en todas las configuraciones posibles.

Así como logramos un buen control con un controlador basado en articulación que estaba basado en un modelo de linealización y desacoplamiento del brazo, también podemos hacer lo mismo para el caso cartesiano. No obstante, ahora debemos escribir las ecuaciones dinámicas de movimiento del manipulador en términos de variables cartesianas, como vimos en el capítulo 6. La forma resultante de las ecuaciones de movimiento es bastante análoga a la versión en espacio de articulación. La dinámica de cuerpo rígido puede escribirse así:

$$\mathcal{F} = M_x(\Theta)\ddot{\chi} + V_x(\Theta, \dot{\Theta}) + G_x(\Theta), \tag{10.53}$$

en donde $\mathcal{F}$ es un vector ficticio de fuerza–momento que actúa sobre el efector final del robot, y χ es un vector cartesiano apropiado que representa la posición y orientación del efector final [8]. Análoga a las cantidades en espacio de articulación, $M_x(\Theta)$ es la matriz de masas en espacio cartesiano, $V_x(\Theta, \dot{\Theta})$ es un vector con términos de velocidad en espacio cartesiano y $G_x(\Theta)$ es un vector con términos de gravedad en el espacio cartesiano.

Al igual que como hicimos en el caso basado en articulación, podemos usar las ecuaciones dinámicas en un controlador de desacoplamiento y linealización. Así como la ecuación (10.53) calcula $\mathcal{F}$, un vector de fuerza cartesiana ficticio que debe aplicarse a la mano, también tendremos que usar la transpuesta del jacobiano para poder implementar el control; esto es, una vez que se calcula $\mathcal{F}$ mediante la ecuación (10.53) no podemos hacer que se aplique una fuerza cartesiana al efector final; en vez de ello calculamos los momentos de torsión de articulación requeridos para balancear efectivamente el sistema si fuéramos a aplicar esta fuerza:

$$\tau = J^T(\Theta)\mathcal{F}. \tag{10.54}$$

La figura 10.14 muestra un sistema de control de brazo cartesiano que utiliza un desacoplamiento dinámico completo. Observe que el brazo va precedido por la transpuesta del jacobiano. Tome en cuenta que el controlador de la figura 10.14 permite describir rutas cartesianas directamente, sin necesidad de conversión de trayectorias.

Al igual que en el caso del espacio de articulación, podría lograrse mejor una implementación práctica a través del uso de un sistema de control de velocidad dual. La figura 10.15 muestra un diagrama de bloques de un controlador de desacoplamiento y linealización de base cartesiana, en el que los parámetros dinámicos se escriben como funciones de la posición del manipulador solamente. Estos parámetros dinámicos se

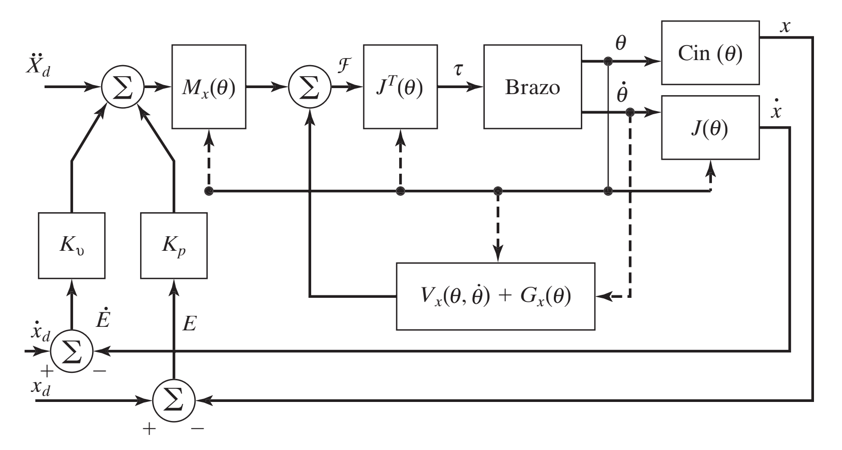


FIGURA 10.14: Esquema de control basado en un modelo cartesiano.

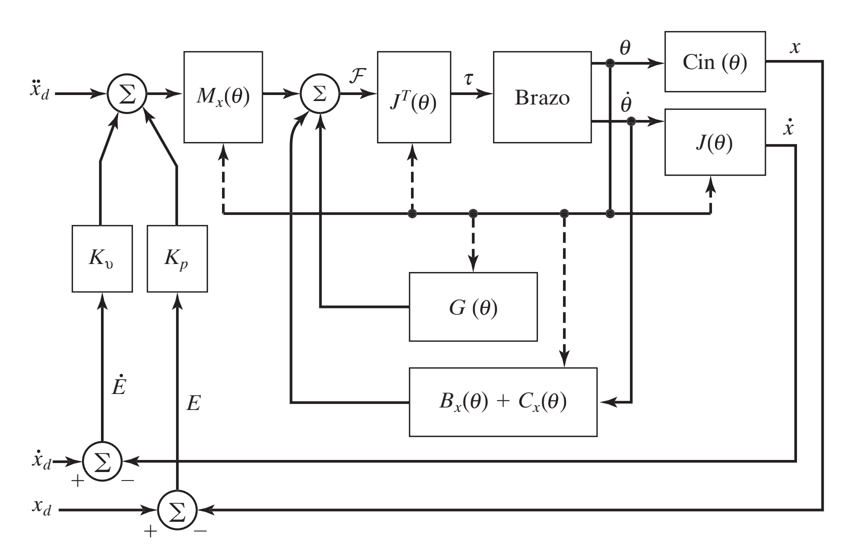


FIGURA 10.15: Una implementación del esquema de control basado en un modelo cartesiano.

actualizan a una velocidad menor que la velocidad del servo, mediante un proceso en segundo plano o una segunda computadora de control, lo cual es apropiado, ya que deseamos un servo veloz (que funcione tal vez a 500 Hz o aún más) para maximizar el rechazo de perturbaciones y la estabilidad. Los parámetros dinámicos son sólo funciones de la posición del manipulador, por lo que necesitan actualizarse a una proporción que se relaciona sólo con la rapidez con la que el manipulador cambia de configuración. La velocidad de actualización de parámetros probablemente no necesita ser mayor de 100 Hz [8].

10.9 CONTROL ADAPTATIVO

Al hablar sobre el control basado en el modelo se recalcó que, a menudo, los parámetros del manipulador no se conocen con exactitud. Cuando los parámetros en el modelo

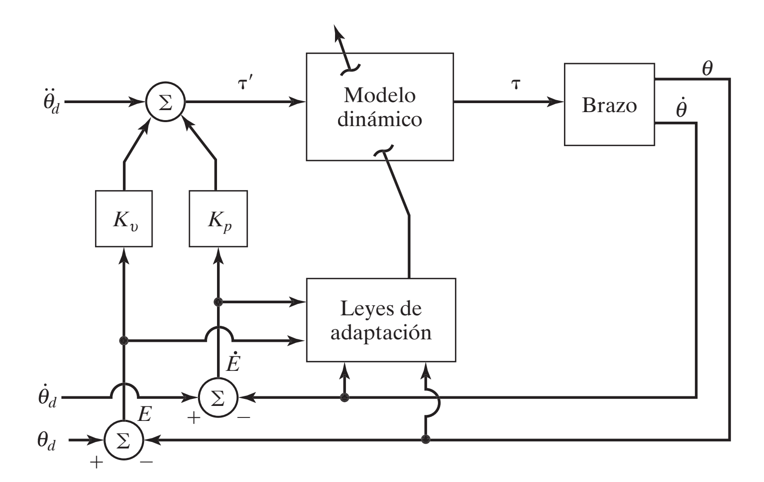


FIGURA 10.16: El concepto de un controlador de manipulador adaptativo.

no concuerdan con los parámetros del dispositivo real se producen errores de servo, como se hace explícito en la ecuación (10.25). Estos errores de servo podrían usarse para controlar cierto esquema de adaptación que intente actualizar los valores de los parámetros del modelo hasta que los errores desaparezcan. Se han propuesto varios esquemas adaptativos de ese tipo.

Un esquema ideal adaptativo podría ser como el de la figura 10.16, donde usamos una ley de control basada en el modelo, como se desarrolló en este capítulo. Hay un proceso de adaptación que, dadas las observaciones del estado del manipulador y los errores de servo, reajusta los parámetros en el modelo no lineal hasta que desaparezcan los errores. Un sistema de ese tipo *aprendería* sus propias propiedades dinámicas. El diseño y el análisis de los esquemas adaptativos es un tema que está más allá del alcance de este libro. Un método que posee la misma estructura que se muestra en la figura 10.16, y que se ha probado que es globalmente estable, se presenta en las referencias [20] y [21]. La referencia [22] utiliza una técnica relacionada.

BIBLIOGRAFÍA

[1] R. P. Paul, “Modeling, Trajectory Calculation, and Servoing of a Computer Controlled Arm”, reporte técnico AIM-177, Laboratorio de Inteligencia Artificial de la Universidad de Stanford, 1972.

[2] B. Markiewicz, “Analysis of the Computed Torque Drive Method and Comparison with Conventional Position Servo for a Computer-Controlled Manipulator”, memorándum técnico 33-601, Jet Propulsion Laboratory, marzo de 1973.

[3] A. Bejczy, “Robot Arm Dynamics and Control”, memorándum técnico 33-669, Jet Propulsion Laboratory, febrero de 1974.

[4] J. LaSalle y S. Lefschetz, *Stability by Liapunov’s Direct Method with Applications*, Academic Press, Nueva York, 1961.

[5] P. K. Koshla, “Some Experimental Results on Model-Based Control Schemes”, *IEEE Conference on Robotics and Automation*, Filadelfia, abril de 1988.

[6] M. Leahy, K. Valavanis y G. Saridis, “The Effects of Dynamic Models on Robot Control”, *IEEE Conference on Robotics and Automation*, San Francisco, abril de 1986.

[7] L. Sciavicco y B. Siciliano, *Modelling and Control of Robot Manipulators*, 2a. edición, Springer-Verlag, Londres, 2000.

[8] O. Khatib, "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation", *IEEE Journal of Robotics and Automation*, Vol. RA-3, núm. 1, 1987.

[9] C. An, C. Atkeson y J. Hollerbach, "Model-Based Control of a Direct Drive Arm, Part II: Control", *IEEE Conference on Robotics and Automation*, Filadelfia, abril de 1988.

[10] G. Franklin, J. Powell y M. Workman, *Digital Control of Dynamic Systems*, 2a. edición, Addison-Wesley, Reading, MA, 1989.

[11] A. Liegeois, A. Fournier y M. Aldon, "Model Reference Control of High Velocity Industrial Robots", *Proceedings of the Joint Automatic Control Conference,* San Francisco, 1980.

[12] M. Raibert, "Mechanical Arm Control Using a State Space Memory," SME paper MS77-750, 1977.

[13] B. Armstrong, "Friction: Experimental Determination, Modeling and Compensation", IEEE Conference on Robotics and Automation, Filadelfia, abril de 1988.

[14] B. Armstrong, O. Khatib y J. Burdick, "The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm", *IEEE Conference on Robotics and Automation*, San Francisco, abril de 1986.

[15] A. M. Lyapunov, "Sobre el Problema General de la Estabilidad de Movimiento" (en ruso), Sociedad Matemática de Kharkov, Unión Soviética, 1982.

[16] C. Desoer y M. Vidyasagar, *Feedback Systems: Input–Output Properties*, Academic Press, Nueva York, 1975.

[17] M. Vidyasagar, *Nonlinear Systems Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1978.

[18] S. Arimoto y F. Miyazaki, "Stability and Robustness of PID Feedback Control for Robot Manipulators of Sensory Capability", *Third International Simposium of Robotics Research*, Gouvieux, Francia, julio de 1985.

[19] D. Koditschek, "Adaptive Strategies for the Control of Natural Motion", *Proceedings of the 24th Conference on Decision and Control*, Ft. Lauderdale, FL, diciembre de 1985.

[20] J. Craig, P. Hsu y S. Sastry, "Adaptive Control of Mechanical Manipulators", *IEEE Conference on Robotics and Automation*, San Francisco, abril de 1986.

[21] J. Craig, *Adaptive Control of Mechanical Manipulators*, Addison-Wesley, Reading, MA, 1988.

[22] J.J. Slotine y W. Li, "On the Adaptive Control of Mechanical Manipulators", *The International Journal of Robotics Research*,Vol. 6, núm. 3, 1987.

[23] R. Kelly y R. Ortega, "Adaptive Control of Robot Manipulators: An Input—Output Approach", *IEEE Conference on Robotics and Automation*, Filadelfia, 1988.

[24] H. Das, J. J. Slotine y T. Sheridan, "Inverse Kinematic Algorithms for Redundant Systems", *IEEE Conference on Robotics and Automation*, Filadelfia, 1988.

[25] T. Yabuta, A. Chona y G. Beni, "On the Asymptotic Stability of the Hybrid Position/Force Control Scheme for Robot Manipulators", *IEEE Conference on Robotics and Automation*, Filadelfia, 1988.

EJERCICIOS

10.1 [15] Encuentre las ecuaciones de control no lineales para un controlador α, β particionado para el sistema

$$\tau = (2\sqrt{\theta} + 1)\ddot{\theta} + 3\dot{\theta}^2 - \operatorname{sen}(\theta).$$

Seleccione las ganancias de forma que este sistema siempre esté críticamente amortiguado con $k_{CL} = 10$.

10.2 [15] Encuentre las ecuaciones de control no lineales para un controlador α, β particionado para el sistema

$$\tau = 5\theta\dot{\theta} + 2\ddot{\theta} - 13\dot{\theta}^3 + 5.$$

Seleccione las ganancias de forma que este sistema siempre esté críticamente amortiguado con $k_{CL} = 10$.

10.3 [19] Dibuje un diagrama de bloque que muestre un controlador en espacio de articulación para el brazo de dos vínculos de la sección 6.7, de tal forma que el brazo esté críticamente amortiguado sobre todo su espacio de trabajo. Muestre las ecuaciones dentro de los bloques de un diagrama de bloques.

10.4 [20] Dibuje un diagrama de bloques que muestre un controlador en espacio cartesiano para el brazo de dos vínculos de la sección 6.7, de tal forma que el brazo esté críticamente amortiguado sobre todo su espacio de trabajo (vea el ejemplo 6.6). Muestre las ecuaciones dentro de los bloques de un diagrama de bloques.

10.5 [18] Diseñe un sistema de control de seguimiento de trayectorias para el sistema cuya dinámica se da mediante las ecuaciones

$$\tau_1 = m_1 l_1^2 \ddot{\theta}_1 + m_1 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2,$$

$$\tau_2 = m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + \upsilon_2 \dot{\theta}_2.$$

¿Cree usted que estas ecuaciones podrían representar un sistema real?

10.6 [17] Para el sistema de control diseñado para el manipulador de un vínculo del ejemplo 10.3, encuentre una expresión para el error de posición de estado estable como una función del error en el parámetro de la masa. Haga que $\psi_m = m - \hat{m}$. El resultado deberá ser una función de l, g, θ, ψ_m, $\hat{m}$ y k_p. ¿Para qué posición del manipulador se obtiene un máximo?

10.7 [26] Para el sistema mecánico de dos grados de libertad de la figura 10.17, diseñe un controlador que pueda hacer que x_1 y x_2 sigan las trayectorias y supriman las perturbaciones en una forma críticamente amortiguada.

10.8 [30] Considere las ecuaciones dinámicas del manipulador de dos vínculos de la sección 6.7, en forma de espacio de configuración. Derive expresiones para la sensibilidad del valor de momento de torsión calculado respecto a pequeñas desviaciones en Θ. ¿Puede decir algo sobre qué tan frecuentemente debe recalcularse el comportamiento dinámico en un controlador como el de la figura 10.7, como una función de las velocidades de articulación promedio esperadas durante las operaciones normales?

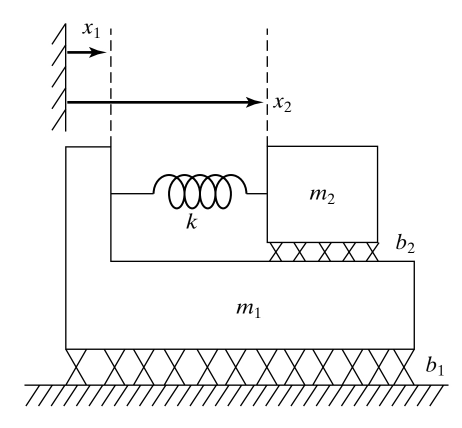


FIGURA 10.17: Sistema mecánico con dos grados de libertad.

10.9 [32] Considere las ecuaciones dinámicas del manipulador de dos vínculos del ejemplo 6.6 en forma de espacio de configuración cartesiano. Derive expresiones para la sensibilidad del valor de momento de torsión calculado respecto a pequeñas desviaciones en Θ. ¿Puede decir algo sobre qué tan frecuentemente debe recalcularse el comportamiento dinámico en un controlador como el de la figura 10.15, como una función de las velocidades de articulación promedio esperadas durante las operaciones normales?

10.10 [15] Diseñe un sistema de control para el sistema

$$f = 5x\dot{x} + 2\ddot{x} - 12.$$

Seleccione las ganancias de manera que este sistema siempre esté críticamente amortiguado, con una rigidez de lazo cerrado de 20.

10.11 [20] Considere un sistema de regulación de posición que (sin perder la generalidad) intente mantener $\Theta_d = 0$. Demuestre que la ley de control

$$\tau = -K_p\Theta - M(\Theta)K_\upsilon\dot{\Theta} + G(\Theta)$$

produce un sistema no lineal asintóticamente estable. Debe tomar K_υ para que sea de la forma $K_\upsilon = k_\upsilon I_n$, en donde k_υ es un escalar e I_n es la matriz identidad de $n \times n$. *Sugerencia*: Esto es similar al ejemplo 10.6.

10.12 [20] Considere un sistema de regulación de posición que (sin perder la generalidad) intente mantener $\Theta_d = 0$. Demuestre que la ley de control

$$\tau = -K_p\Theta - \hat{M}(\Theta)K_\upsilon\dot{\Theta} + G(\Theta)$$

produce un sistema no lineal asintóticamente estable. Debe tomar K_υ para que sea de la forma $K_\upsilon = k_\upsilon I_n$, en donde k_υ es un escalar e I_n es la matriz identidad de $n \times n$. La matriz $\hat{M}(\Theta)$ es una estimación positiva definida de la matriz de masas del manipulador. *Sugerencia*: Esto es similar al ejemplo 10.6.

10.13 [25] Considere un sistema de regulación de posición que (sin perder la generalidad) intente mantener $\Theta_d = 0$. Demuestre que la ley de control

$$\tau = -M(\Theta)[K_p\Theta + K_\upsilon\dot{\Theta}] + G(\Theta)$$

produce un sistema no lineal asintóticamente estable. Debe tomar K_υ para que sea de la forma $K_\upsilon = k_\upsilon I_n$, en donde k_υ es un escalar e I_n es la matriz identidad de $n \times n$. *Sugerencia*: Esto es similar al ejemplo 10.6.

10.14 [25] Considere un sistema de regulación de posición que (sin perder la generalidad) intente mantener $\Theta_d = 0$. Demuestre que la ley de control

$$\tau = -\hat{M}(\Theta)[K_p\Theta + K_\upsilon\dot{\Theta}] + G(\Theta)$$

produce un sistema no lineal asintóticamente estable. Debe tomar K_υ para que sea de la forma $K_\upsilon = k_\upsilon I_n$, en donde k_υ es un escalar e I_n es la matriz de identidad $n \times n$. La matriz $\hat{M}(\Theta)$ es una estimación positiva definida de la matriz de masas del manipulador. *Sugerencia*: Esto es similar al ejemplo 10.6.

10.15 [28] Considere un sistema de regulación de posición que (sin perder la generalidad) intente mantener $\Theta_d = 0$. Demuestre que la ley de control

$$\tau = -K_p\Theta - K_\upsilon\dot{\Theta}$$

produce un sistema no lineal estable. Demuestre que la estabilidad no es asintótica y dé una expresión para el error de estado estable. *Sugerencia*: Esto es similar al ejemplo 10.6.

10.16 [30] Demuestre la estabilidad global del controlador cartesiano de jacobiano transpuesto que se presentó en la sección 10.8. Use una forma apropiada de retroalimentación de velocidad para estabilizar el sistema.
Sugerencia: Vea la referencia [18].

10.17 [15] Diseñe un controlador de seguimiento de trayectorias para un sistema con una dinámica dada por

$$f = ax^2\dot{x}\ddot{x} + b\dot{x}^2 + c\,\mathrm{sen}(x),$$

de tal forma que los errores se supriman en forma críticamente amortiguada sobre todas las configuraciones.

10.18 [15] Un sistema con una dinámica de lazo abierto dada por

$$\tau = m\ddot{\theta} + b\dot{\theta}^2 + c\dot{\theta}$$

se controla mediante la ley de control

$$\tau = m[\ddot{\theta}_d + k_v\dot{e} + k_p e] + \mathrm{sen}(\theta).$$

Encuentre la ecuación diferencial que caracterice la acción de lazo cerrado del sistema.

EJERCICIO DE PROGRAMACIÓN (PARTE 10)

Repita el ejercicio de programación de la parte 9 y utilice las mismas pruebas, pero con un nuevo controlador que utilice un modelo dinámico completo de los 3 vínculos para desacoplar y linealizar el sistema. Para este caso, utilice

$$K_p = \begin{bmatrix} 100.0 & 0.0 & 0.0 \\ 0.0 & 100.0 & 0.0 \\ 0.0 & 0.0 & 100.0 \end{bmatrix}.$$

Seleccione una K_v diagonal que garantice un amortiguamiento crítico sobre todas las configuraciones del brazo. Compare los resultados con los que se obtuvieron con el controlador más simple utilizado en el ejercicio de programación, parte 9.

C A P Í T U L O 11

Control de fuerza de los manipuladores

11.1 INTRODUCCIÓN

El control de la posición es apropiado cuando un manipulador está siguiendo una trayectoria por el espacio, pero cuando se hace cualquier tipo de contacto entre el efector final y el ambiente del manipulador, el simple control de la posición no es suficiente. Considere un manipulador que lava una ventana con una esponja. La elasticidad de la esponja podría hacer posible la regulación de la fuerza que se aplica a la ventana, mediante el control de la posición del efector final con relación al vidrio. Si la esponja es muy elástica o si la posición del vidrio se conoce con mucha precisión, esta técnica podría funcionar bastante bien.

No obstante, si la rigidez del efector final, de la herramienta o del entorno es alta, se vuelve muy difícil realizar operaciones en las que el manipulador ejerza presión contra una superficie. En vez de lavar con una esponja, imagine que el manipulador está raspando pintura de una superficie de vidrio usando una herramienta de raspado rígida. Si hay incertidumbre respecto a la posición de la superficie del vidrio, o si existe algún error en la posición del manipulador, esta tarea se volvería imposible. O se rompería el vidrio, o el manipulador agitaría la herramienta de raspado sobre el vidrio sin que hubiera ningún tipo de contacto.

En las tareas de lavado y de raspado sería más razonable no especificar la posición del plano del vidrio, sino *especificar una fuerza que se va a mantener normal a la superficie*.

En este capítulo, más que en los anteriores, presentaremos métodos que aún no son empleados por robots industriales, excepto de una manera extremadamente simplificada. El concepto más importante del capítulo es introducir el **controlador híbrido de posición/fuerza**, el cual es un formalismo mediante el cual los robots industriales podrían controlarse algún día para realizar tareas que requieran del control de la fuerza. No obstante, sin importar cuál o cuáles métodos emerjan como prácticos para la aplicación industrial, muchos de los conceptos presentados en este capítulo seguirán siendo ciertamente válidos.

11.2 APLICACIÓN DE ROBOTS INDUSTRIALES A TAREAS DE ENSAMBLAJE

La mayoría de los robots industriales se emplea en **aplicaciones simples** (relativamente hablando), tales como soldadura por puntos pintura con pistola de aire y operaciones de levantamiento y colocación. El control de la fuerza ya ha aparecido en algunas aplicaciones; por ejemplo, algunos robots ya cuentan con un control de fuerza simple que les permite realizar tareas tales como triturado y eliminación de rebabas. Aparentemente la siguiente área de aplicación será en las tareas de líneas de ensamblaje en las que se unen una o más piezas. En dichas tareas de **unión de piezas** es extremadamente importante el monitoreo y el control de las fuerzas de contacto.

El control preciso de los manipuladores para enfrentar incertidumbres y variaciones en sus entornos de trabajo es el prerrequisito para la aplicación de manipuladores de robots en las operaciones de ensamblaje en la industria. Al parecer, si se colocan sensores en las manos de un manipulador para que proporcionen información sobre el estado de las tareas que realizan, habría un gran avance hacia el uso de robots para las tareas de ensamblaje. Actualmente la destreza de los manipuladores sigue siendo considerablemente baja y continúa limitando su aplicación en el área de ensamblaje automatizado.

Para utilizar manipuladores en tareas de ensamblaje se requiere que la precisión con la que las piezas se colocan, una respecto a la otra, sea bastante alta. En general, los robots industriales actuales no son lo suficiente precisos para estas tareas, y podría no tener sentido construir robots que pudieran serlo. Pueden construirse manipuladores de una mayor precisión pero a expensas del tamaño, peso y costo. Sin embargo, la habilidad de medir y controlar las fuerzas de contacto generadas en la mano ofrece una posible alternativa para extender la precisión efectiva de una manipulación. Como se utilizan mediciones relativas, los errores absolutos en la posición del manipulador y en los objetos manipulados no son tan importantes como lo serían en un sistema controlado solamente por la posición. Las pequeñas variaciones en la posición relativa generan grandes fuerzas de contacto cuando interactúan piezas de rigidez moderada, por lo que el conocimiento y el control de estas fuerzas puede producir un tremendo incremento en la precisión posicional efectiva.

11.3 UN PARADIGMA DE CONTROL EN TAREAS PARCIALMENTE RESTRINGIDAS

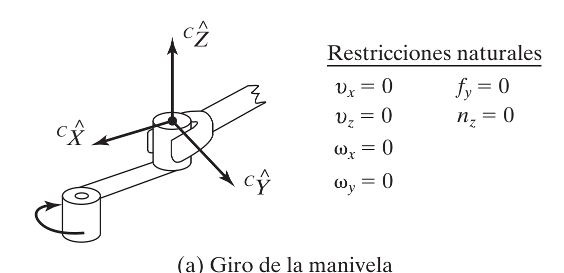
El método presentado en este capítulo se basa en un marco de trabajo para el control en situaciones en las que el movimiento del manipulador está parcialmente restringido por el contacto con una o más superficies [1-3]. Este marco de trabajo para entender las tareas parcialmente restringidas se basa en un modelo simplificado de interacción entre el efector final del manipulador y el entorno: nos interesa describir el contacto y las libertades, por lo que sólo consideramos las fuerzas debido al contacto. Esto es equivalente a realizar un análisis casi estático e ignorar otras fuerzas estáticas, como ciertos componentes de fricción y de gravedad. El análisis es razonable cuando las fuerzas debidas al contacto entre objetos relativamente rígidos son la fuente dominante de fuerzas que actúan en el sistema. Observe que la metodología que se presenta aquí es algo simplista y tiene ciertas limitaciones, pero es una buena forma de introducir los conceptos básicos implicados y lo hace a un nivel apropiado para este texto. Para una metodología relacionada pero más general y rigurosa, vea la referencia [19].

Toda tarea de manipulación puede descomponerse en subtareas que se definen mediante una situación de contacto específica que ocurre entre el efector final del manipulador (o herramienta) y el entorno de trabajo. Con cada subtarea podemos asociar un conjunto de restricciones, llamadas **restricciones naturales**, que se producen debido

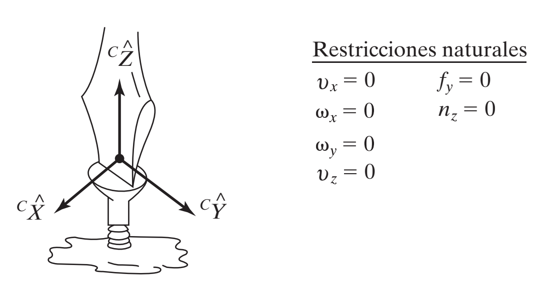
a las características mecánicas y geométricas específicas de la configuración de la tarea. Por ejemplo, una mano en contacto con una superficie estacionaria y rígida no está libre para moverse a través de esa superficie; por ende, existe una restricción de posición *natural*. Si la superficie no tiene fricción, la mano no está libre para aplicar fuerzas arbitrarias tangentes a la superficie; por ende, existe una restricción de fuerza *natural*.

En nuestro modelo de contacto con el entorno, para la configuración de cada subtarea puede definirse una **superficie generalizada** con restricciones de posición a lo largo de las normales para esta superficie, y restricciones de fuerza a lo largo de las tangentes. Estos dos tipos de restricción (fuerza y posición) particionan los grados de libertad de los posibles movimientos del efector final en dos conjuntos ortogonales que deben controlarse de acuerdo con varios criterios. Observe que este modelo de contacto no incluye todas las posibles situaciones de contacto. (Vea la referencia [19] para un esquema más general).

La figura 11.1 muestra dos tareas representativas junto con sus restricciones naturales asociadas. Observe que, en cada caso, la tarea se describe en términos de una trama $\{C\}$, la llamada **trama de restricción**, que se posiciona en una ubicación relevante para la tarea. De acuerdo con la tarea, $\{C\}$ podría estar fija en el entorno o podría moverse con el efector final del manipulador. En la figura 11.1(a), la trama de restricción está unida a la manivela como se muestra y se mueve junto con ella, con la dirección $\hat{X}$ indicando siempre hacia el punto pivote de la manivela. La fricción que actúa en la punta de los dedos asegura una sujeción firme sobre el mango, el cual está sobre un eje de manera que pueda girar con respecto al brazo manivela. En la figura 11.1(b) la trama de restricción está unida a la punta del destornillador y se mueve con él a medida que se desarrolla la tarea. Observe que en la dirección $\hat{Y}$ la fuerza se restringe para ser cero, ya que la ranura del tornillo permitiría al destornillador deslizarse en esa dirección. En estos ejemplos se sigue cumpliendo un conjunto dado de restricciones durante toda la tarea. En situaciones más complejas, la tarea se descompone en subtareas para las que pueda identificarse un conjunto constante de restricciones naturales.



(a) Giro de la manivela



(b) Giro del destornillador

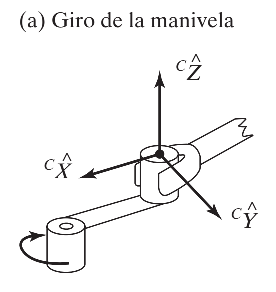
FIGURA 11.1: Las restricciones naturales para dos tareas distintas.

En la figura 11.1 se han indicado las restricciones de posición dando valores para los componentes de velocidad del efector final V, descrito en la trama $\{C\}$. Podríamos haber indicado las restricciones de posición dando expresiones para la posición en vez de velocidades; no obstante, en muchos casos es más simple especificar una restricción de posición como una restricción de "velocidad igual a cero". De igual forma se han especificado las restricciones de fuerza dando valores para los componentes del vector de fuerza-momento $\mathcal{F}$, que actúa sobre el efector final descrito en la trama $\{C\}$. Observe que cuando decimos *restricciones de posición* indicamos restricciones de posición o de orientación, y cuando decimos *restricciones de fuerza* indicamos restricciones de fuerza o de momento. El término *restricciones naturales* se utiliza para indicar que estas restricciones surgen naturalmente de la situación de contacto específica. No tienen nada que ver con el movimiento deseado o esperado del manipulador.

Las restricciones adicionales, llamadas **restricciones artificiales**, se introducen de acuerdo con las restricciones naturales para especificar movimientos deseados o la aplicación de fuerza. Esto es, cada vez que el usuario especifica una trayectoria deseada, ya sea en posición o en fuerza, se define una restricción artificial. Estas restricciones también ocurren a lo largo de las tangentes y las normales de la superficie de restricción generalizada pero, a diferencia de las restricciones naturales, las restricciones de fuerza artificiales se especifican a lo largo de las normales de la superficie, y las restricciones de posición artificiales se especifican a lo largo de las tangentes; por ende, se preserva la consistencia con las restricciones naturales.

La figura 11.2 muestra las restricciones naturales y artificiales para dos tareas. Observe que cuando se da una restricción de posición natural para un cierto grado de

(a) Giro de la manivela



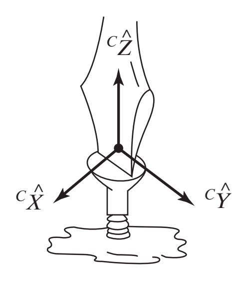
Restricciones naturales

$\upsilon_x = 0$ $\quad f_y = 0$
$\upsilon_z = 0$ $\quad n_z = 0$
$\omega_x = 0$
$\omega_y = 0$

Restricciones artificiales

$\upsilon_y = r\alpha_1$ $\quad f_x = 0$
$\omega_z = \alpha_1$ $\quad f_z = 0$
$n_x = 0$
$n_y = 0$

(b) Giro del destornillador



Restricciones naturales

$\upsilon_x = 0$ $\quad f_y = 0$
$\omega_x = 0$ $\quad n_z = 0$
$\omega_y = 0$
${}_z = 0$

Restricciones artificiales

$\upsilon_y = 0$ $\quad f_x = 0$
$\omega_z = \alpha_2$ $\quad n_x = 0$
$n_y = 0$
$f_z = \alpha_3$

FIGURA 11.2: Las restricciones naturales y artificiales para dos tareas.

libertad en $\{C\}$ debe especificarse una restricción de fuerza artificial, y viceversa. En cualquier instante, cualquier grado de libertad en la trama de restricción se controla para cumplir con una restricción de posición o de fuerza.

Estrategia de ensamblaje es un término que hace referencia a una secuencia de restricciones artificiales planeadas que hacen que la tarea proceda en una manera deseable. Dichas estrategias deben incluir métodos mediante los cuales el sistema pueda detectar un cambio en la situación de contacto para que puedan rastrearse las transiciones en las restricciones naturales. Con cada cambio de ese tipo en las restricciones naturales, se recupera un nuevo conjunto de restricciones artificiales del conjunto de estrategias de ensamblaje para que el sistema de control las aplique. Los métodos para elegir automáticamente las restricciones para una tarea de ensamblaje dada requieren de más investigación. En este capítulo vamos a suponer que se ha analizado una tarea para poder determinar las restricciones naturales y que un planeador humano ha determinado una **estrategia de ensamblaje** con la cual se va a controlar el manipulador.

Hay que tener en cuenta que por lo general ignoraremos las fuerzas de fricción entre las superficies de contacto en nuestro análisis de las tareas. Esto bastará para nuestra introducción al problema y, de hecho, producirá estrategias que funcionarán en muchos casos. Por lo general las fuerzas de fricción de deslizamiento actúan en direcciones elegidas para ser controladas en posición, y por lo tanto estas fuerzas aparecen como perturbaciones para el servo de posición y son superadas por el sistema de control.

EJEMPLO 11.1

La figura 11.3(a)-(d) muestra una secuencia de ensamblaje utilizada para colocar una chaveta redonda en un hoyo redondo. La chaveta se baja hacia la superficie a la izquierda del hoyo y luego se desliza sobre la superficie hasta que cae en el hoyo. Después se inserta hasta que la chaveta llega hasta el fondo del hoyo, en donde el ensamblaje se considera completo. Cada una de las cuatro situaciones de contacto indicadas define una subtarea. Proporcione las restricciones naturales y artificiales para cada una de las subtareas mostradas. Además, indique cómo el subsistema detecta el cambio en las restricciones naturales a medida que procede la operación.

Primero uniremos la trama de restricción a la chaveta, como se muestra en la figura 11.3(a). En esta figura la chaveta está en espacio libre, por lo que las restricciones naturales son

$$^{C}\mathcal{F} = 0. \tag{11.1}$$

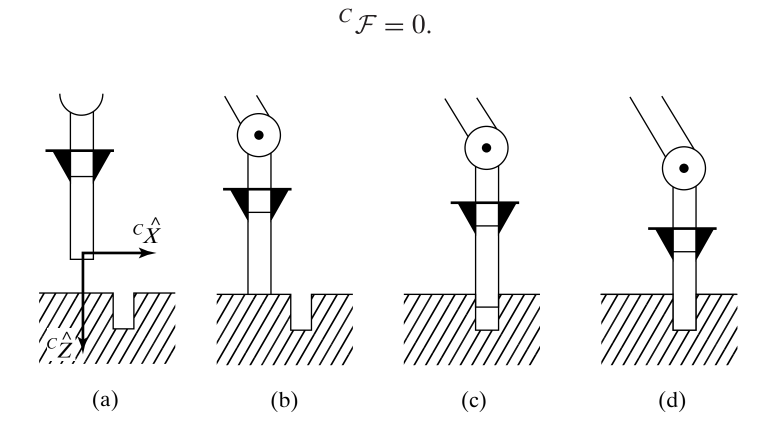


FIGURA 11.3: La secuencia de cuatro situaciones de contacto para insertar una chaveta.

En este caso, las restricciones artificiales constituyen una trayectoria de posición completa, la cual mueve a la chaveta en la dirección ${}^C\hat{Z}$ hacia la superficie. Por ejemplo,

$$ {}^C\nu = \begin{bmatrix} 0 \\ 0 \\ \upsilon_{\text{acercamiento}} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{11.2} $$

en donde $\upsilon_{\text{acercamiento}}$ es la velocidad con la cual se acerca a la superficie.

En la figura 11.3(b) la chaveta ha llegado a la superficie. Para detectar que esto ha ocurrido, observamos la fuerza en la dirección ${}^C\hat{Z}$. Cuando esta fuerza detectada pasa de un umbral determinamos el contacto, lo cual implica una nueva situación de contacto con un nuevo conjunto de restricciones naturales. Suponiendo que la situación de contacto es como se muestra en la figura 11.3(b), la chaveta no tiene la libertad de moverse en ${}^C\hat{Z}$ ni de girar sobre ${}^C\hat{X}$ o ${}^C\hat{Y}$. En los otros tres grados no tiene la libertad de aplicar fuerzas; por lo tanto, las restricciones naturales son

$$ \begin{aligned} {}^C\upsilon_z &= 0, \\ {}^C\omega_x &= 0, \\ {}^C\omega_y &= 0, \\ {}^Cf_x &= 0, \\ {}^Cf_y &= 0, \\ {}^Cn_z &= 0. \end{aligned} \tag{11.3} $$

Las restricciones artificiales describen la estrategia para deslizarse a lo largo de la superficie en la dirección ${}^C\hat{X}$ mientras se aplican fuerzas pequeñas para asegurar que se mantenga el contacto. Por lo tanto, tenemos que

$$ \begin{aligned} {}^C\upsilon_x &= \upsilon_{\text{deslizamiento}}, \\ {}^C\upsilon_y &= 0, \\ {}^C\omega_z &= 0, \\ {}^Cf_z &= f_{\text{contacto}}, \\ {}^Cn_x &= 0, \\ {}^Cn_y &= 0. \end{aligned} \tag{11.4} $$

en donde f_{contacto} es la fuerza que se aplica normal a la superficie a medida que se desliza la chaveta, y $\upsilon_{\text{deslizamiento}}$ es la velocidad con la cual se desliza a través de la superficie.

En la figura 11.3(c) la chaveta ha caído ligeramente dentro del hoyo. Esta situación se detecta observando la velocidad en la dirección ${}^C\hat{Z}$ y esperando a que cruce un umbral (que se haga distinta de cero, en el caso ideal). Cuando esto ocurre, nos indica que una vez más han cambiado las restricciones naturales, y por lo tanto nuestra estrategia (como

se expresa en las restricciones artificiales) debe cambiar. Las nuevas restricciones naturales son

$$
\begin{aligned}
{}^{C}\upsilon_x &= 0,\\
{}^{C}\upsilon_y &= 0,\\
{}^{C}\omega_x &= 0,\\
{}^{C}\omega_y &= 0,\\
{}^{C}f_x &= 0,\\
{}^{C}n_z &= 0.
\end{aligned}
\tag{11.5}
$$

Elegimos las restricciones artificiales para que sean

$$
\begin{aligned}
{}^{C}\upsilon_z &= \upsilon_{\text{inserción}},\\
{}^{C}\omega_z &= 0,\\
{}^{C}f_x &= 0,\\
{}^{C}f_y &= 0,\\
{}^{C}n_x &= 0,\\
{}^{C}n_y &= 0,
\end{aligned}
\tag{11.6}
$$

en donde $\upsilon_{\text{inserción}}$ es la velocidad a la que se inserta la chaveta en el hoyo. Finalmente, la situación que se muestra en la figura 11.3(d) se detecta cuando la fuerza en la dirección ${}^{C}\hat{Z}$ aumenta por encima de un umbral.

Es interesante observar que los cambios en las restricciones naturales siempre se detectan observando la variable de posición o de fuerza que *no* se está controlando. Por ejemplo, para detectar la transición de la figura 11.3(b) a la figura 11.3(c), monitoreamos la velocidad en ${}^{C}\hat{Z}$ mientras controlamos la fuerza en ${}^{C}\hat{Z}$. Para saber cuándo llega la chaveta al fondo del hoyo, monitoreamos ${}^{C}f_z$, aunque estamos controlando ${}^{C}\upsilon_z$.

El marco de trabajo que hemos introducido es algo simplista. En la referencia [19] se podrá encontrar un método más general y riguroso de “dividir” las tareas en componentes controlados por posición y componentes controlados por fuerza.

La tarea de determinar estrategias de ensamblaje para ajustar piezas más complicadas entre sí es considerablemente compleja. También hemos ignorado los efectos de la incertidumbre en nuestro sencillo análisis de esta tarea. El desarrollo de sistemas de planeación automática que incluyen los efectos de la incertidumbre y pueden aplicarse a situaciones prácticas ha sido un tema de investigación [4-8] desde hace tiempo. Para una buena revisión de estos métodos consulte la referencia [9].

11.4 EL PROBLEMA DE CONTROL HÍBRIDO DE POSICIÓN/FUERZA

La figura 11.4 muestra dos ejemplos extremos de situaciones de contacto. En la figura 11.4(a), el manipulador se está moviendo a través del espacio libre. En este caso, las restricciones naturales son todas de fuerza; no hay nada contra qué reaccionar, por lo

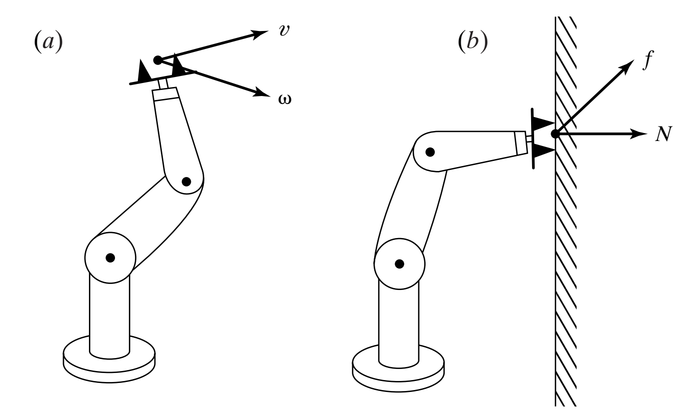


FIGURA 11.4: Los dos extremos en situaciones de contacto. El manipulador a la izquierda se mueve en el espacio libre, en donde no existe una superficie de reacción. El manipulador a la derecha está pegado a la pared, por lo que no es posible el movimiento libre.

que todas las fuerzas se restringen para hacerse cero.[1] Con un brazo que tiene seis grados de libertad, podemos movernos en seis grados de libertad en posición pero no es posible ejercer fuerzas en ninguna dirección. La figura 11.4(b) muestra la situación extrema de un manipulador con su efector final pegado a una pared. En este caso, el manipulador está sujeto a seis restricciones naturales de posición ya que no está libre para reposicionarse. No obstante, el manipulador está libre para ejercer fuerzas y momentos de torsión sobre el objeto con seis grados de libertad.

En los capítulos 9 y 10 estudiamos el problema de control de posición que se aplica a la situación de la figura 11.4(a). La situación de la figura 11.4(b) no ocurre muy a menudo en la práctica; generalmente debemos considerar el control de la fuerza en el contexto de tareas parcialmente restringidas en las que ciertos grados de libertad del sistema están sujetos al control de posición y otros están sujetos al control de fuerza. Por lo tanto, en este capítulo consideraremos los esquemas **híbridos de control de posición/fuerza**.

El controlador híbrido de posición/fuerza debe resolver tres problemas:

1. El control de la posición de un manipulador sobre las direcciones en las que exista una restricción natural de fuerza.
2. El control de la fuerza de un manipulador sobre las direcciones en las que exista una restricción natural de posición.
3. Un esquema para implementar la mezcla arbitraria de esos modos sobre los grados de libertad ortogonales de una trama arbitraria $\{C\}$.

11.5 CONTROL DE FUERZA DE UN SISTEMA DE MASA-RESORTE

En el capítulo 9 comenzamos nuestro estudio del problema completo de control de posición con el estudio del problema bastante simple de controlar un solo bloque de masa. Después, en el capítulo 10 pudimos utilizar un modelo del manipulador de tal forma que el problema de controlar todo el manipulador se hizo equivalente a controlar n masas

[1] Es importante recordar que aquí nos concentramos en las *fuerzas de contacto* entre el efector final y el entorno, no en las fuerzas inerciales.

independientes (para un manipulador con n articulaciones). De manera similar, comenzamos nuestro análisis del control de fuerza controlando la fuerza aplicada por un sistema simple con un solo grado de libertad.

Al considerar las fuerzas de contacto debemos hacer un modelo del entorno sobre el cual estamos actuando. Para los fines del desarrollo, conceptual utilizaremos un modelo muy simple de interacción entre un cuerpo controlado y el entorno. Modelaremos el contacto con un entorno como un resorte; esto es, supondremos que nuestro sistema es rígido y que el entorno tiene cierta rigidez, la cual está representada por k_e.

Consideremos el control de una masa unida a un resorte, como en la figura 11.5. También incluiremos una fuerza de perturbación desconocida, representada por f_{pert}, la cual podría considerarse como un modelo para la fricción desconocida o los dientes en el engranaje del manipulador. La variable que deseamos controlar es la fuerza que actúa en el entorno, f_e, que es la fuerza que actúa en el resorte:

$$f_e = k_e x. \tag{11.7}$$

La ecuación que describe este sistema físico es

$$f = m\ddot{x} + k_e x + f_{\text{pert}}, \tag{11.8}$$

o, si se escribe en términos de la variable que deseamos controlar, f_e,

$$f = mk_e^{-1}\ddot{f}_e + f_e + f_{\text{pert}}. \tag{11.9}$$

Usando el concepto de controlador particionado, así como

$$\alpha = mk_e^{-1}$$

y

$$\beta = f_e + f_{\text{pert}}$$

llegamos a la ley de control,

$$f = mk_e^{-1}\left[\ddot{f}_d + k_{vf}\dot{e}_f + k_{pf}e_f\right] + f_e + f_{\text{pert}}, \tag{11.10}$$

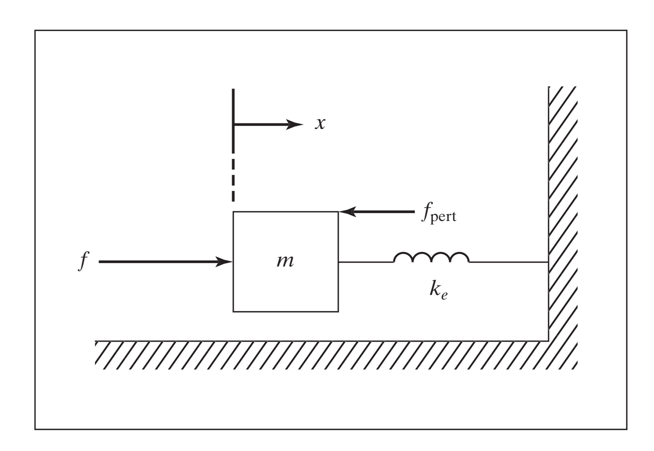


FIGURA 11.5: Un sistema de resorte-masa.

en donde $e_f = f_d - f_e$ es el error de fuerza entre la fuerza deseada, f_d, y la fuerza detectada en el entorno, f_e. Si pudiéramos calcular la ecuación (11.10), tendríamos el sistema de lazo cerrado

$$\ddot{e}_f + k_{vf}\dot{e}_f + k_{pf}e_f = 0. \tag{11.11}$$

Sin embargo, no podemos utilizar el conocimiento de f_{pert} en nuestra ley de control, por lo que la ecuación (11.10) no es viable. Podríamos dejar ese término fuera de la ley de control, pero un análisis de estado estable muestra que hay una mejor opción, especialmente cuando la rigidez del entorno (k_e) es alta (la situación usual).

Si optamos por dejar el término f_{pert} fuera de nuestra ley de control, igualamos las ecuaciones (11.9) y (11.10) y hacemos un análisis de estado estable haciendo que todas las derivadas en el tiempo sean cero, tendremos que

$$e_f = \frac{f_{\text{pert}}}{\alpha}, \tag{11.12}$$

en donde $\alpha = mk_e^{-1}k_{pf}$, la ganancia de retroalimentación de fuerza efectiva; no obstante, si optamos por usar f_d en la ley de control (11.10) en vez del término $f_e + f_{\text{pert}}$, encontraremos que el error de estado estable será

$$e_f = \frac{f_{\text{dist}}}{1+\alpha}. \tag{11.13}$$

Cuando el entorno es rígido (como es común en este caso) α podría ser pequeña, de tal forma que el error calculado en la ecuación (11.13) es una mejora considerable respecto al de la ecuación (11.12). Por lo tanto, sugerimos la siguiente ley de control:

$$f = mk_e^{-1}\left[\ddot{f}_d + k_{vf}\dot{e}_f + k_{pf}e_f\right] + f_d. \tag{11.14}$$

La figura 11.6 es un diagrama de bloques del sistema de lazo cerrado que utiliza la ley de control (11.14).

En general, las consideraciones prácticas cambian la implementación de un servo de control de fuerza de manera considerable, a diferencia del ideal que se muestra en la figura 11.6. En primer lugar, las trayectorias de fuerza son generalmente constantes;

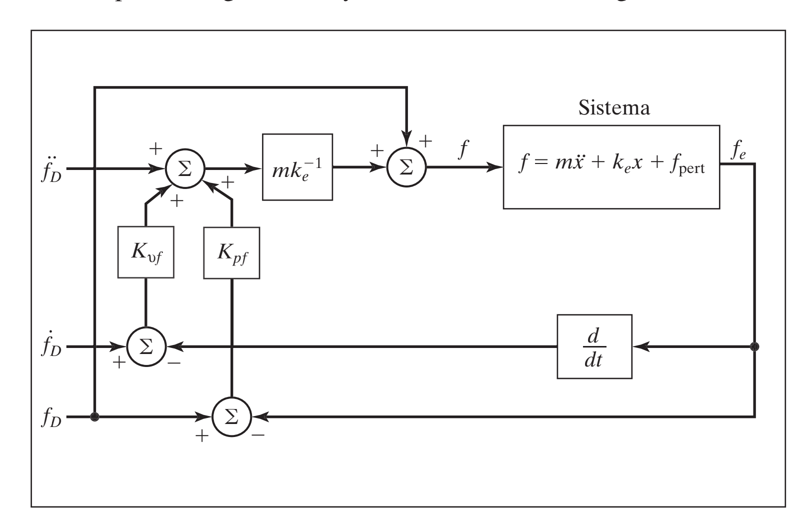


FIGURA 11.6: Un sistema de control de fuerza para el sistema de resorte-masa.

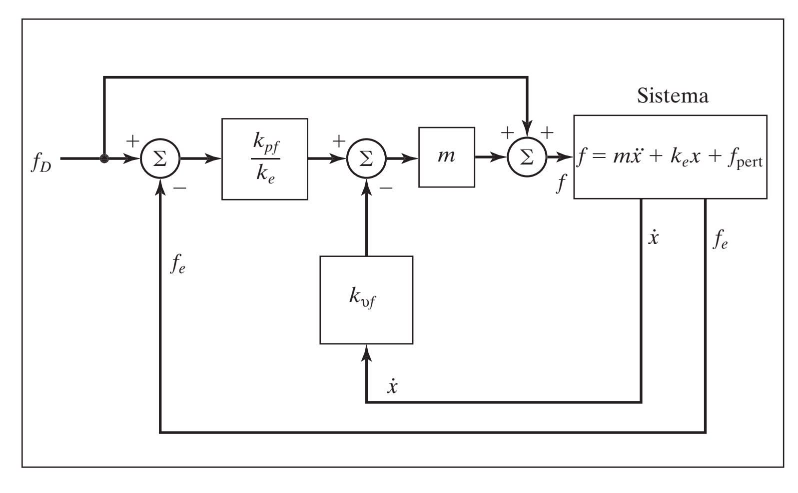


FIGURA 11.7: Un sistema práctico de control de fuerza para el sistema de resorte-masa.

esto es, por lo general nos interesa controlar la fuerza de contacto para que esté a cierto nivel constante. Son raras las aplicaciones en las que las fuerzas de contacto deban seguir cierta función arbitraria de tiempo. Por lo tanto, las entradas $\dot{f}_d$ y $\ddot{f}_d$ del sistema de control muy a menudo se establecen permanentemente en cero. Otra realidad es que las fuerzas detectadas son bastante "ruidosas", por lo que sería imprudente usar la diferenciación numérica para calcular $\dot{f}_e$. No obstante, $f_e = k_e x$, por lo que podemos obtener la derivada de la fuerza en el entorno como $\dot{f}_e = k_e \dot{x}$. Esto es mucho más realista ya que la mayoría de los manipuladores tiene los medios para obtener buenas medidas de velocidad. Después de haber hecho estas dos elecciones pragmáticas, escribimos la ley de control como

$$f = m\left[k_{pf} k_e^{-1} e_f - k_{vf}\dot{x}\right] + f_d, \tag{11.15}$$

con el correspondiente diagrama de bloques que se muestra en la figura 11.7.

Considere que una interpretación del sistema de la figura 11.7 es que los errores de fuerza generan un punto de ajuste para un lazo interno de control de velocidad con una ganancia k_{vf}. Algunas leyes de control de fuerza también incluyen un término integral para mejorar el rendimiento en estado estable.

Un importante problema remanente es que la rigidez del entorno k_e aparece en nuestra ley de control, pero a menudo se desconoce y tal vez cambia ocasionalmente. Sin embargo, con mucha frecuencia un robot de ensamblaje trata con piezas rígidas, por lo que podría pensarse que k_e sería bastante alta. Generalmente se hace esta suposición, y las ganancias se seleccionan de tal forma que el sistema sea algo robusto respecto a variaciones en k_e.

El propósito de construir una ley para controlar la fuerza de contacto ha sido el de mostrar una estructura que ha sido sugerida y exponer algunos puntos importantes. Durante el resto de este capítulo simplemente supondremos que podría construirse dicho servo de control de fuerza y abstraerlo en una caja negra, como se muestra en la figura 11.8. En la práctica no es fácil construir un servo de fuerza de alto rendimiento, lo que es en la actualidad un área de mucha investigación [11-14]. Para una buena revisión de este tema consulte la referencia [15].

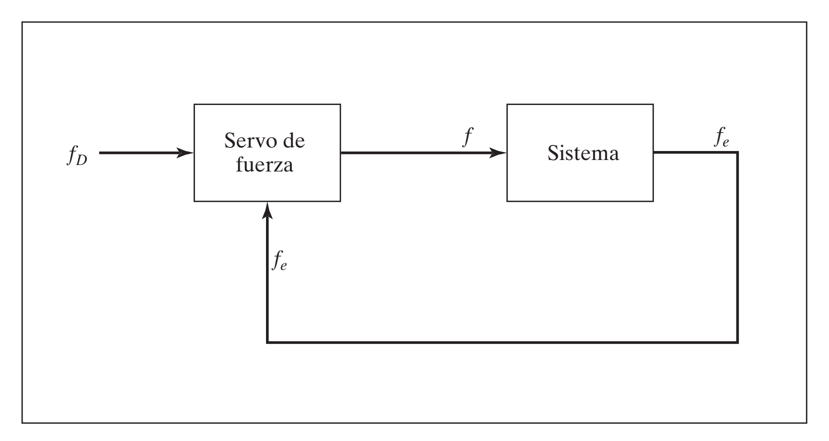


FIGURA 11.8: El servo de control de fuerza como una caja negra.

11.6 EL ESQUEMA DE CONTROL HÍBRIDO DE POSICIÓN/FUERZA

En esta sección presentaremos una arquitectura para un sistema de control que implementa el controlador híbrido de posición/fuerza.

Un manipulador cartesiano alineado con {C}

Primero consideraremos el caso simple de un manipulador con tres grados de libertad y articulaciones prismáticas que actúa en las direcciones $\hat{Z}$, $\hat{Y}$ y $\hat{X}$. Por sencillez supondremos que cada vínculo tiene una masa m y se desliza sobre cojinetes libres de fricción. Vamos a suponer también que los movimientos de las articulaciones se alinean exactamente con la trama de restricción $\{C\}$. El efector final está en contacto con una superficie de rigidez k_e orientada con su normal en la dirección $-{}^C\hat{Y}$. Por lo tanto se requiere el control de fuerza en esa dirección, pero el control de posición en las direcciones ${}^C\hat{X}$ y ${}^C\hat{Z}$. (Vea la figura 11.9).

En este caso, la solución al problema del control híbrido de posición/fuerza es clara. Debemos controlar las articulaciones 1 y 3 con el controlador de posición que desarrollamos para una masa unitaria en el capítulo 9. La articulación 2 (que opera en la dirección $\hat{Y}$) debe manejarse con el controlador de fuerza que desarrollamos en la sección 11.4. Podríamos entonces suministrar una trayectoria de posiciones en las direcciones ${}^C\hat{X}$ y ${}^C\hat{Z}$, y al mismo tiempo podríamos suministrar una trayectoria de fuerzas (tal vez sólo una constante) en la dirección ${}^C\hat{Y}$.

Si deseamos poder cambiar la naturaleza de la superficie de restricción de tal forma que su normal pueda también ser $\hat{X}$ o $\hat{Z}$, podemos generalizar ligeramente nuestro sistema de control de brazo cartesiano de la siguiente manera: construiremos la estructura del controlador de tal forma que podamos especificar una trayectoria de posiciones completa en los tres grados de libertad, y también una trayectoria de fuerzas en esos mismos tres grados. Desde luego no podemos controlar esto de manera que se cumplan estas seis restricciones en un momento dado; pero sí podemos establecer modos de indicar qué componentes de qué trayectoria se seguirán en cualquier momento dado.

Considere el controlador que se muestra en la figura 11.10. Aquí indicamos el control de las tres articulaciones de nuestro brazo cartesiano simple en un diagrama sencillo, mostrando tanto el controlador de posición como el controlador de fuerza. Se han introducido las matrices S y S' para definir el modo (posición o fuerza) que se va a utilizar para controlar cada articulación del brazo cartesiano. La matriz S es diagonal, con unos y ceros en la diagonal.

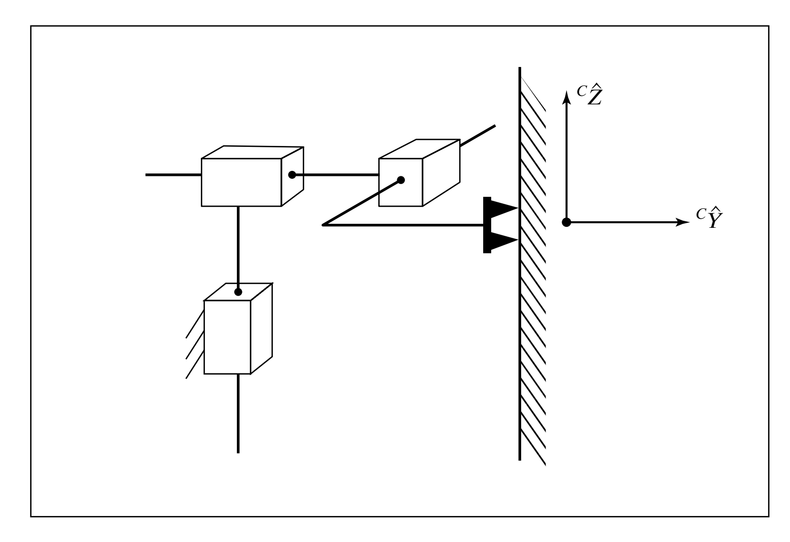


FIGURA 11.9: Un manipulador cartesiano con tres grados de libertad en contacto con una superficie.

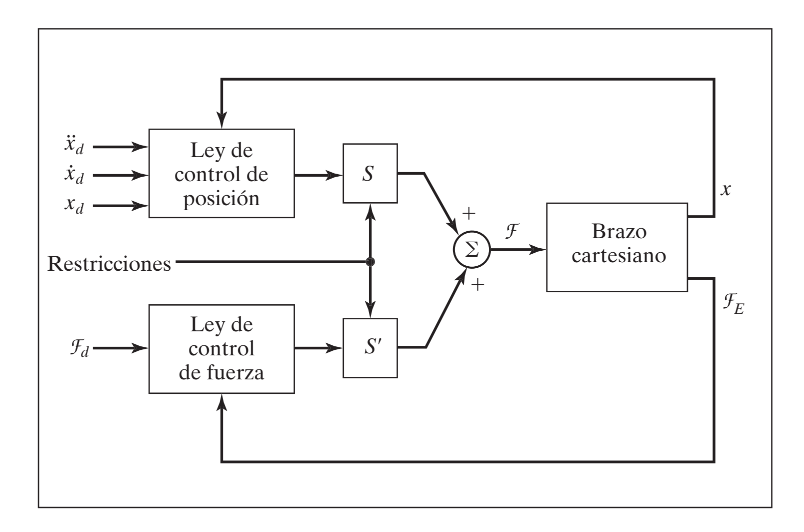


FIGURA 11.10: El controlador híbrido para un brazo cartesiano 3-DOF.

Cuando hay un uno presente en S, hay un cero presente en S' y el control de posición está en efecto. Cuando hay un cero presente en S, hay un uno presente en S' y el control de fuerza está en efecto. Por ende, las matrices S y S' son simplemente interruptores que establecen el modo de control a utilizar con cada grado de libertad en $\{C\}$. De acuerdo con la configuración de S, siempre hay tres componentes de la trayectoria que se están controlando, aunque la mezcla entre control de posición y de fuerza es arbitraria. Se ignoran los otros tres componentes de trayectoria deseada y errores de servo asociados; en consecuencia, cuando cierto grado de libertad está bajo el control de fuerza se ignoran los errores de posición en ese grado de libertad.

EJEMPLO 11.2

Encuentre las matrices S y S' para la situación que se muestra en la figura 11.9, con los movimientos en la dirección ${}^{C}\hat{Y}$ restringidos por la superficie de reacción.

Como los componentes $\hat{X}$ y $\hat{Z}$ se van a controlar por posición, introducimos unos en la diagonal de S que sean correspondientes a estos dos componentes. Esto hará que el servo de posición se active en estas dos direcciones y se seguirá la trayectoria de entrada; se ignorará cualquier entrada de trayectoria de posiciones para el componente $\hat{Y}$. La matriz S' tiene los unos y ceros invertidos en la diagonal; por lo tanto, tenemos

$$S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$S' = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{11.16}$$

La figura 11.10 muestra el controlador híbrido para el caso especial en que las articulaciones se alinean exactamente con la trama de restricción $\{C\}$. En la siguiente subsección utilizaremos las técnicas estudiadas en capítulos anteriores para generalizar el controlador para trabajar con manipuladores generales y para una $\{C\}$ arbitraria. No obstante, en el caso ideal el sistema funciona como si el manipulador tuviera un actuador "alineado" con cada grado de libertad en $\{C\}$.

Un manipulador general

El proceso de generalizar el controlador híbrido mostrado en la figura 11.10 de manera que pueda usarse un manipulador general es sencillo con el concepto del control de base cartesiana. En el capítulo 6 vimos cómo escribir las ecuaciones de movimiento de un manipulador en términos del movimiento cartesiano del efector final, y en el capítulo 10 vimos cómo utilizar dicha formulación para lograr un control de posición cartesiano desacoplado de un manipulador. La idea principal es que, a través del uso de un modelo dinámico escrito en espacio cartesiano, es posible controlar de manera tal que el sistema combinado del manipulador actual y del modelo calculado aparezcan como un conjunto de masas unitarias desacopladas e independientes. Una vez realizados estos procesos de desacoplamiento y linealización, podemos aplicar el servo simple desarrollado en la sección 11.4.

La figura 11.11 muestra la compensación basada en la formulación de la dinámica del manipulador en espacio cartesiano, de tal forma que el manipulador aparezca como un conjunto de masas unitarias desacopladas. Para usarse en el esquema de control híbrido, la dinámica cartesiana y el jacobiano se escriben en la trama de restricción $\{C\}$. De igual forma, la cinemática se calcula respecto a la trama de restricción.

Como hemos diseñado el controlador híbrido para un manipulador cartesiano alineado con la trama de restricción, y como el esquema de desacoplamiento cartesiano nos proporciona un sistema con las mismas propiedades de entrada–salida, sólo necesitamos combinarlos para generar el controlador híbrido generalizado de posición/fuerza.

La figura 11.12 es un diagrama de bloques del controlador híbrido para un manipulador general. Observe que la dinámica está escrita en la trama de restricción, al igual que el jacobiano.

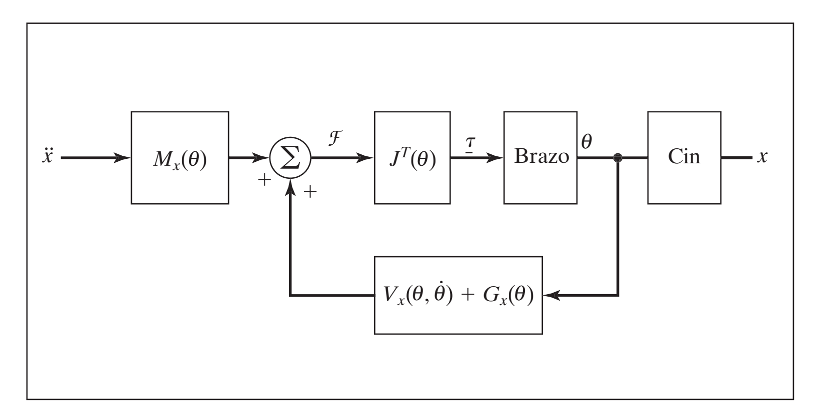


FIGURA 11.11: El esquema de desacoplamiento cartesiano introducido en el capítulo 10.

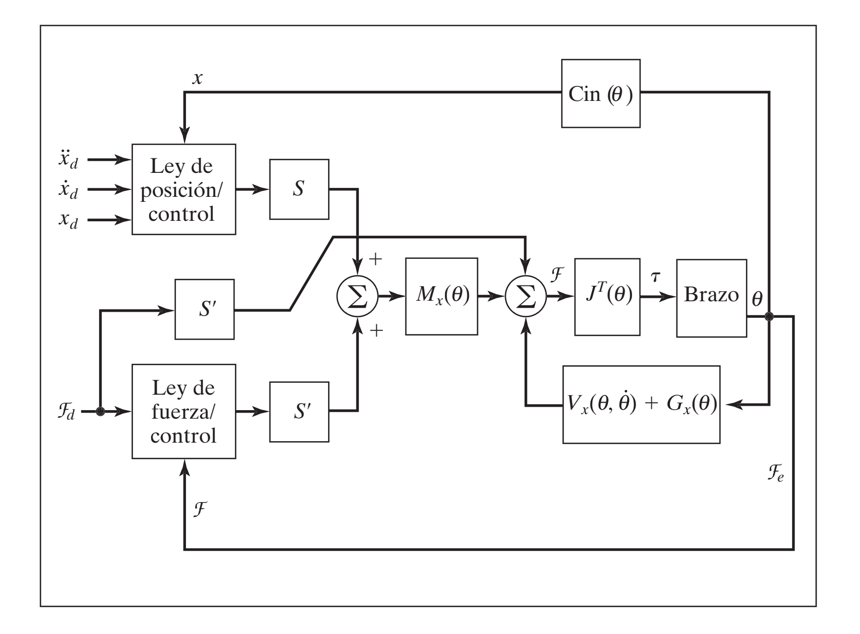


FIGURA 11.12: El controlador híbrido de posición/fuerza para un manipulador general. Por simplicidad no se ha mostrado el lazo de velocidad-retroalimentación.

Las ecuaciones cinemáticas están escritas para incluir la transformación de coordenadas en la trama de restricción, y las fuerzas detectadas se transforman de igual forma en $\{C\}$. Los errores de servo se calculan en $\{C\}$ y los modos de control dentro de $\{C\}$ se establecen a través de la elección apropiada de S.[2] La figura 11.13 muestra un manipulador controlado por un sistema de este tipo.

[2] La partición de los modos de control a lo largo de ciertas direcciones relacionadas con la tarea se ha generalizado en [10], con base en el método más básico que se presentó en este capítulo.

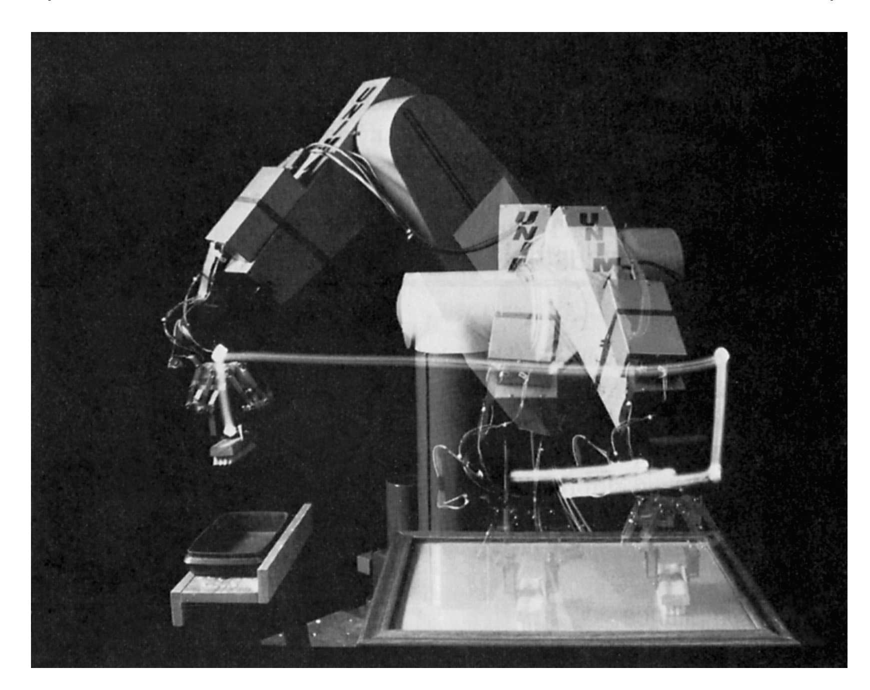


FIGURA 11.13: Un manipulador PUMA 560 lava una ventana bajo el control del sistema COSMOS desarrollado bajo la responsabilidad de O. Khatib en la Universidad de Stanford. Estos experimentos utilizan dedos detectores de fuerzas y una estructura de control similar a la de la figura 11.12 [10].

Cómo agregar rigidez variable

La acción de controlar un grado de libertad en un estricto control de posición o de fuerza representa el control en dos extremos del espectro de rigidez del servo. Un servo de posición ideal es infinitamente rígido y rechaza todas las perturbaciones de fuerza que actúan en el sistema. En contraste, un servo de fuerza ideal exhibe cero rigidez y mantiene la aplicación de fuerza deseada sin importar las perturbaciones de posición. Podría ser útil controlar el efector final para exhibir rigideces distintas de cero o infinito. En general, podría ser conveniente controlar la **impedancia mecánica** del efector final [14, 16, 17].

En nuestro análisis del contacto hemos imaginado que el entorno es muy rígido. Cuando hacemos contacto con un entorno rígido, utilizamos un control de fuerza con cero rigidez; y cuando hacemos contacto con cero rigidez (movimiento en espacio libre), utilizamos un control de posición con alta rigidez. Por ende, parece que controlar el efector final para exhibir una rigidez que sea aproximadamente la inversa del entorno local es una buena estrategia. En consecuencia, al tratar con piezas de plástico o resortes podría ser conveniente establecer una rigidez del servo distinta de cero o infinito.

Dentro del marco de trabajo del controlador híbrido, esto se hace simplemente utilizando el control de posición y reduciendo la ganancia de posición que corresponde al grado apropiado de libertad en $\{C\}$. En general, al hacer eso se reduce la ganancia de velocidad de manera que el grado de libertad siga siendo críticamente amortiguado. La habilidad de cambiar las ganancias de posición y de velocidad del servo de posición

a lo largo de los grados de libertad de $\{C\}$ permite al controlador híbrido de posición/fuerza implementar una impedancia generalizada del efector final [17]. No obstante, en muchas situaciones prácticas estamos tratando con la intersección de piezas rígidas de manera que se desea el control puro de posición o el control puro de fuerza.

11.7 ESQUEMAS DE CONTROL DE ROBOTS INDUSTRIALES EN LA ACTUALIDAD

El verdadero control de fuerza, como el controlador híbrido de posición/fuerza que presentamos en este capítulo, no existe actualmente en los robots industriales. Entre los problemas de implementación práctica están la enorme cantidad de poder de cómputo requerido, la falta de parámetros precisos para el modelo dinámico, la falta de sensores de fuerza de uso rudo y el nivel de dificultad impuesto al usuario para especificar una estrategia de posición/fuerza.

Adaptabilidad pasiva

Los manipuladores extremadamente rígidos con servos de posición muy rígidos son inadecuados para las tareas en las que las piezas entran en contacto y se generan fuerzas de contacto. En tales situaciones es frecuente que las piezas se atoren o se dañen. Desde los primeros experimentos con los manipuladores que intentaban realizar ensamblajes se dedujo que, en la medida en que los robots podían realizar dichas tareas, se debía a la adaptabilidad de las piezas, o de los dispositivos fijos, o del brazo en sí. Esta habilidad de una o más piezas del sistema de "ceder" un poco era a menudo suficiente para permitir la unión exitosa de las piezas.

Una vez que se dedujo lo anterior, se diseñaron dispositivos especiales para introducir a propósito adaptabilidad en el sistema. El dispositivo más exitoso de ese tipo es el RCC o dispositivo de *adaptabilidad de centro remoto*, desarrollado en Draper Labs [18]. El RCC fue diseñado inteligentemente de manera que introdujera el tipo "correcto" de adaptabilidad, lo que permitía que ciertas tareas se llevaran a cabo sin problemas y rápidamente con poca o ninguna probabilidad de atascamientos. El RCC es básicamente un resorte con seis grados de libertad que se inserta entre la muñeca del manipulador y el efector final. Al ajustar las rigideces de los seis resortes pueden introducirse diversos niveles de adaptabilidad. Dichos esquemas se conocen como esquemas de **adaptabilidad pasiva** y se utilizan en aplicaciones industriales de manipuladores en ciertas tareas.

Adaptabilidad mediante la suavización de las ganancias de posición

En lugar de lograr la adaptabilidad en una forma pasiva, y por lo tanto fija, es posible crear esquemas en los que la aparente rigidez del manipulador se altere a través del ajuste de las ganancias de un sistema de control de posición. Algunos robots industriales hacen algo parecido para aplicaciones tales como el triturado, en donde es necesario mantener contacto con una superficie pero no se requiere el control delicado de la fuerza.

Salisbury [16] ha sugerido un método especialmente interesante. En este esquema, las ganancias de posición en un sistema de servo basado en articulación se modifican de forma tal que el efector final parezca tener cierta rigidez a lo largo de los grados cartesianos de libertad: considere un resorte general con seis grados de libertad. Su acción podría describirse por

$$\mathcal{F} = K_{px}\delta\chi, \tag{11.17}$$

en donde K_{px} es una matriz diagonal de 6×6 con tres rigideces lineales seguidas por tres rigideces torsionales en la diagonal. ¿Cómo podríamos hacer que el efector final de un manipulador exhibiera esta característica de rigidez?

Recordando la definición del jacobiano del manipulador, tenemos que

$$\delta\chi = J(\Theta)\delta\Theta. \tag{11.18}$$

Si la combinamos con la ecuación (11.17) obtenemos

$$\mathcal{F} = K_{px} J(\Theta)\delta\Theta. \tag{11.19}$$

De las consideraciones de fuerza estática tenemos que

$$\tau = J^T(\Theta)\mathcal{F}, \tag{11.20}$$

lo que, combinado con la ecuación (11.19), nos produce

$$\tau = J^T(\Theta) K_{px} J(\Theta)\delta\Theta. \tag{11.21}$$

Aquí el jacobiano se escribe generalmente en la trama herramienta. La ecuación (11.21) es una expresión para la forma en que deben generarse los momentos de torsión de una articulación como una función de pequeños cambios en los ángulos de articulación $\delta\Theta$, para poder hacer que el efector final del manipulador se comporte como un resorte cartesiano con seis grados de libertad.

Mientras que un simple controlador de posición basado en articulación podría utilizar la ley de control

$$\tau = K_p E + K_v \dot{E}, \tag{11.22}$$

en donde K_p y K_v son matrices diagonales de ganancias constantes y E es el error de servo definido como $\Theta_d - \Theta$, Salisbury sugiere utilizar

$$\tau = J^T(\Theta) K_{px} J(\Theta) E + K_v \dot{E}, \tag{11.23}$$

en donde K_{px} es la rigidez deseada del efector final en espacio cartesiano. Para un manipulador con seis grados de libertad, K_{px} es una matriz diagonal con los seis valores de la diagonal que representan las tres rigideces traslacionales y las tres rigideces de rotación que va a exhibir el efector final. Básicamente, mediante el uso del jacobiano se ha transformado una rigidez cartesiana en una rigidez en el espacio de articulación.

Detección de fuerza

La **detección de fuerza** permite a un manipulador detectar el contacto con una superficie y, utilizando esta sensación, realizar alguna acción. Por ejemplo, el término **movimiento protegido** se utiliza algunas veces para indicar la estrategia de moverse bajo el control de posición hasta que se detecta una fuerza para después detener el movimiento. Además, la detección de fuerza puede utilizarse como una verificación simple durante una operación de manejo de piezas para asegurar que se haya adquirido una pieza o que se haya adquirido la pieza apropiada.

Algunos robots disponibles comercialmente vienen equipados con sensores de fuerza en el efector final. Estos robots pueden programarse para detener el movimiento

o tomar cualquier otra acción cuando se excede un umbral de fuerza, y algunos pueden ser programados para pesar objetos que se sujeten en el efector final.

BIBLIOGRAFÍA

[1] M. Mason, "Compliance and Force Control for Computer Controlled Manipulators", tesis de maestría, Laboratorio de Inteligencia Artificial (IA) del MIT, mayo de 1978.

[2] J. Craig y M. Raibert, "A Systematic Method for Hybrid Position/Force Control of a Manipulator", *Proceedings of the 1979 IEEE Computer Software Applications Conference*, Chicago, noviembre de 1979.

[3] M. Raibert y J. Craig, "Hybrid Position/Force Control of Manipulators", *ASME Journal of Dynamic Systems, Measurement, and Control*, junio de 1981.

[4] T. Lozano-Perez, M. Mason y R. Taylor, "Automatic Synthesis of Fine-Motion Strategies for Robots", *1st International Symposium of Robotics Research*, Bretton Woods, NH, agosto de 1983.

[5] M. Mason, "Automatic Planning of Fine Motions: Correctness and Completeness", IEEE International Conference on Robotics, Atlanta, marzo de 1984.

[6] M. Erdmann, "Using Backprojections for the Fine Motion Planning with Uncertainty", *The International Journal of Robotics Research*, Vol. 5, núm. 1, 1986.

[7] S. Buckley, "Planning and Teaching Compliant Motion Strategies", disertación doctoral, Departamento de Ingeniería Eléctrica y Ciencias Computacionales, MIT, enero de 1986.

[8] B. Donald, "Error Detection and Recovery for Robot Motion Planning with Uncertainty", disertación doctoral, Departamento de Ingeniería Eléctrica y Ciencias Computacionales, MIT, julio de 1987.

[9] J. C. Latombe, "Motion Planning with Uncertainty: On the Preimage Backchaining Approach", en *The Robotics Review*, O. Khatib, J. Craig y T. Lozano-Perez, editores, MIT Press, Cambridge, MA, 1988.

[10] O. Khatib, "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation", *IEEE Journal of Robotics and Automation*, Vol. RA-3, núm. 1, 1987.

[11] D. Whitney, "Force Feedback Control of Manipulator Fine Motions", *Proceedings Joint Automatic Control Conference*, San Francisco, 1976.

[12] S. Eppinger y W. Seering, "Understanding Bandwidth Limitations in Robot Force Control", *Proceedings of the IEEE Conference on Robotics and Automation*, Raleigh, NC, 1987.

[13] W. Townsend y J.K. Salisbury, "The Effect of Coulomb Friction and Stiction on Force Control", *Proceedings of the IEEE Conference on Robotics and Automation*, Raleigh, NC, 1987.

[14] N. Hogan, "Stable Execution of Contact Tasks Using Impedance Control", *Proceedings of the IEEE Conference on Robotics and Automation*, Raleigh, NC, 1987.

[15] N. Hogan y E. Colgate, "Stability Problems in Contact Tasks", *The Robotics Review*, O. Khatib, J. Craig y T. Lozano-Perez, editores, MIT Press, Cambridge, MA, 1988.

[16] J. K. Salisbury, "Active Stiffness Control of a Manipulator in Cartesian Coordinates", *19th IEEE Conference on Decision and Control*, diciembre de 1980.

[17] J. K. Salisbury y J. Craig, "Articulated Hands: Force Control and Kinematic Issues", *International Journal of Robotics Research*, Vol. 1, no. 1.

[18] S. Drake, "Using Compliance in Lieu of Sensory Feedback for Automatic Assembly", tesis doctoral, Departamento de Ingeniería Mecánica, MIT, septiembre de 1977.

[19] R. Featherstone, S. S. Thiebaut y O. Khatib, "A General Contact Model for Dynamically-Decoupled Force/Motion Control", *Proceedings of the IEEE Conference on Robotics and Automation*, Detroit, 1999.

EJERCICIOS

11.1 [12] Encuentre las restricciones naturales presentes para una chaveta de corte transversal cuadrado que se desliza en un hoyo de corte transversal cuadrado. Muestre su definición de $\{C\}$ en un bosquejo.

11.2 [10] Encuentre las restricciones artificiales (es decir, la trayectoria) que sugeriría para hacer que la chaveta del ejercicio 11.1 se deslice más abajo en el hoyo sin atascarse.

11.3 [20] Demuestre que al usar la ley de control (11.4) con un sistema dado por la ecuación (11.9) se produce la siguiente ecuación en el espacio de error:

$$\ddot{e}_f + k_{v_f}\dot{e}_f + (k_{pf} + m^{-1}k_e)e_f = m^{-1}k_e f_{\text{pert}},$$

y que por consecuencia es posible elegir ganancias para proporcionar un amortiguamiento crítico sólo si se conoce la rigidez del entorno, representada por k_e.

11.4 [17] Dada

$$^{A}_{B}T = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 10.0 \\ 0.500 & 0.866 & 0.000 & 0.0 \\ 0.000 & 0.000 & 1.000 & 5.0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

si el vector de fuerza–momento de torsión en el origen de $\{A\}$ es

$$^{A}\nu = \begin{bmatrix} 0.0 \\ 2.0 \\ -3.0 \\ 0.0 \\ 0.0 \\ 4.0 \end{bmatrix},$$

encuentre el vector de fuerza–momento de torsión de 6×1 con el punto de referencia en el origen de $\{B\}$.

11.5 Dada

$$^{A}_{B}T = \begin{bmatrix} 0.866 & 0.500 & 0.000 & 10.0 \\ -0.500 & 0.866 & 0.000 & 0.0 \\ 0.000 & 0.000 & 1.000 & 5.0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

si el vector de fuerza–momento de torsión en el origen de $\{A\}$ es

$$^{A}\nu = \begin{bmatrix} 6.0 \\ 6.0 \\ 0.0 \\ 5.0 \\ 0.0 \\ 0.0 \end{bmatrix},$$

encuentre el vector de fuerza–momento de torsión de 6×1 con el punto de referencia en el origen de $\{B\}$.

11.6 [18] Describa en español cómo lograría insertar un libro en un hueco estrecho entre los libros de su atestada repisa.

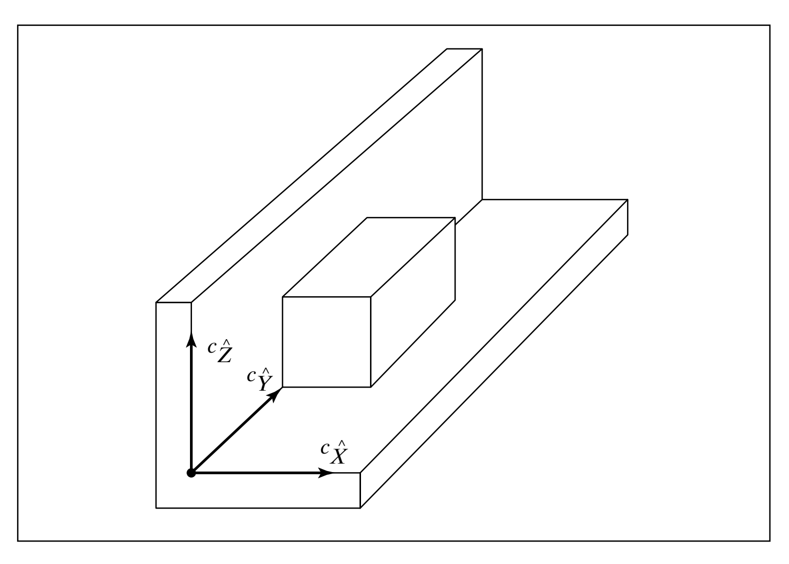


FIGURA 11.14: Un bloque restringido por un piso debajo y una pared al lado.

11.7 [20] Encuentre las restricciones naturales y artificiales para la tarea de cerrar una puerta tipo batiente con un manipulador. Haga todas las suposiciones necesarias. Muestre su definición de $\{C\}$ en un bosquejo.

11.8 [20] Encuentre las restricciones naturales y artificiales para la tarea de descorchar una botella de champaña con un manipulador. Haga todas las suposiciones necesarias. Muestre su definición de $\{C\}$ en un bosquejo.

11.9 [41] Para el sistema de servo de rigidez de la sección 11.7 no hemos declarado que sea estable. Suponga que se utiliza la ecuación (11.23) como la porción de servo de un manipulador desacoplado y linealizado (de manera que las n articulaciones aparezcan como masas unitarias). Demuestre que el controlador es estable para cualquier K_v que sea negativa definida.

11.10 [48] Para el sistema de servo de rigidez de la sección 11.7 no hemos declarado que sea o que pueda ser críticamente amortiguado. Suponga que se utiliza la ecuación (11.23) como la porción de servo de un manipulador desacoplado y linealizado (de manera que las n articulaciones aparezcan como masas unitarias). ¿Es posible diseñar una K_p que sea una función de Θ y haga que el sistema sea críticamente amortiguado sobre todas las configuraciones?

11.11 [15] Como se muestra en la figura 11.14, un bloque está restringido en la parte de abajo por un piso y por la pared en un lado. Suponiendo que esta situación de contacto se mantenga por un tiempo definido, proporcione las restricciones naturales que estén presentes.

EJERCICIO DE PROGRAMACIÓN (PARTE 11)

Implemente un sistema de control de rigidez cartesiano para el manipulador planar de tres vínculos, utilizando la ley de (11.23) para controlar el brazo simulado. Use el jacobiano escrito en la trama {3}.

Para el manipulador en la posición $\Theta = \{60.0, -90.0\ 30.0]$ y para K_{px} de la forma

$$K_{px} = \begin{bmatrix} k_{\text{chica}} & 0.0 & 0.0 \\ 0.0 & k_{\text{grande}} & 0.0 \\ 0.0 & 0.0 & k_{\text{grande}} \end{bmatrix},$$

simule la aplicación de las siguientes fuerzas estáticas:

1. Una fuerza de 1 newton que actúa en el origen de {3} en la dirección $\hat{X}_3$.
2. Una fuerza de 1 newton que actúa en el origen de {3} en la dirección $\hat{Y}_3$.

Los valores de k_{chica} y k_{grande} deben encontrarse en forma experimental. Use un valor grande de k_{grande} para una rigidez alta en la dirección $\hat{Y}_3$ y un valor chico de k_{chica} para una rigidez baja en la dirección $\hat{X}_3$. ¿Cuáles son las deflexiones de estado estable en los dos casos?

C A P Í T U L O 12

Lenguajes y sistemas de programación de robots

12.1 INTRODUCCIÓN

En este capítulo empezaremos a considerar la interfaz entre el usuario humano y un robot industrial. Es mediante esta interfaz que un usuario puede aprovechar todos los algoritmos de mecánica y de control subyacentes que hemos estudiado en capítulos anteriores.

La sofisticación de la interfaz de usuario se ha vuelto en extremo importante a medida que los manipuladores y demás tipos de automatización programable se emplean en cada vez más aplicaciones industriales demandantes. La naturaleza de la interfaz de usuario es una cuestión muy importante; de hecho, la mayor parte del reto del diseño y uso de robots industriales prioriza este aspecto del problema.

Los manipuladores robots se diferencian a sí mismos de la automatización fija al ser “flexibles”, lo cual significa que son programables. No sólo son programables los movimientos de los manipuladores sino que, a través del uso de sensores y comunicación con otros tipos de automatización de fábrica, los manipuladores pueden *adaptarse* a las variaciones a medida que se desarrolle la tarea.

Al considerar la programación de manipuladores es importante recordar que generalmente son sólo una parte mínima de un proceso automatizado. El término **célula de trabajo** se utiliza para describir una colección local de equipo, que puede incluir uno o más manipuladores, sistemas de transportadores, alimentadores de piezas y dispositivos fijos. En el siguiente nivel más alto, las células de trabajo pueden interconectarse en redes distribuidas en toda la fábrica, de manera que una computadora central pueda controlar todo el flujo. Por ende la programación de manipuladores se considera comúnmente dentro del problema más amplio de programar una variedad de máquinas interconectadas en una célula de trabajo de fábrica automatizada.

A diferencia de los 11 capítulos anteriores, el material de este capítulo (y del siguiente) es de una naturaleza que cambia constantemente; por lo tanto, es difícil presentarlo de forma detallada. Trataremos de resaltar los conceptos fundamentales subyacentes y dejaremos al lector la tarea de buscar los ejemplos más recientes.

12.2 LOS TRES NIVELES DE PROGRAMACIÓN DE ROBOTS

A través del tiempo se han desarrollado muchos estilos de interfaz de usuario para programar robots. Antes de la rápida proliferación de microcomputadoras en la industria, los controladores de robots se asemejaban a los secuenciadores simples utilizados a menudo para controlar la automatización fija. Los métodos modernos se enfocan en la programación de computadoras, y las cuestiones relacionadas con la programación de robots incluyen todo lo referente a la programación de computadoras en general; y más.

Enseñar mostrando

Los primeros robots se programaban mediante un método al que llamaremos **enseñar mostrando**, el cual implicaba mover el robot a un punto de destino deseado y registrar su posición en una memoria que el secuenciador leería durante la reproducción. Durante la fase de enseñanza, el usuario guiaba al robot ya sea en forma manual o a través de la interacción con un **dispositivo de enseñanza**. Los dispositivos de enseñanza son cajas de botones portátiles que permiten el control de cada articulación del manipulador o de cada grado de libertad cartesiano. Algunos controladores de este tipo permiten las pruebas y las ramificaciones, por lo que pueden introducirse simples programas que involucren la lógica. Algunos dispositivos de enseñanza tienen pantallas alfanuméricas y se aproximan en complejidad a las terminales portátiles. La figura 12.1 muestra a un operador utilizando un dispositivo de enseñanza para programar un robot industrial.

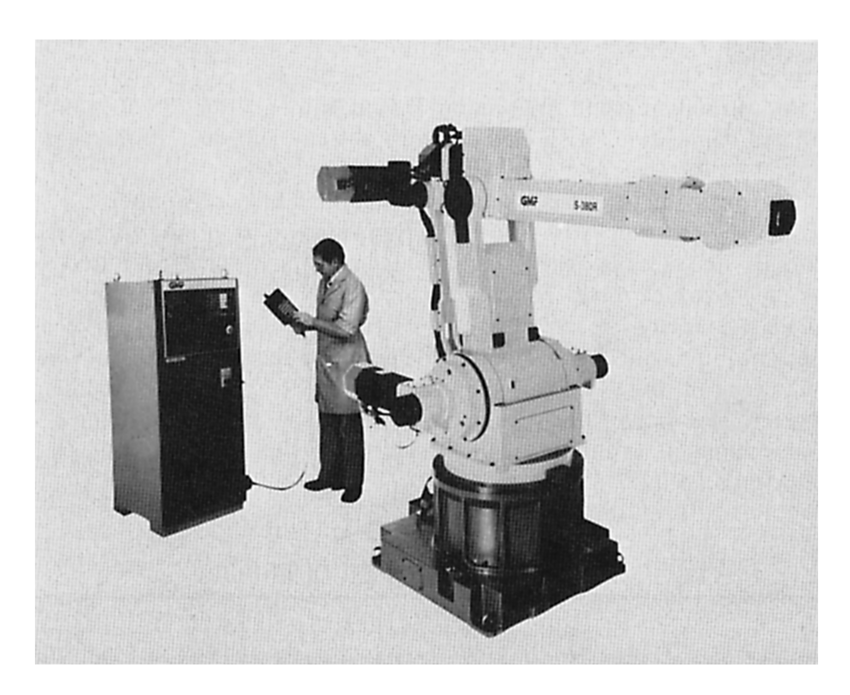


FIGURA 12.1: El GMF 380 se utiliza comúnmente en aplicaciones de soldadura por puntos en carrocerías automotrices. Aquí, un operador utiliza una interfaz de dispositivo de enseñanza para programar el manipulador. Fotografía cortesía de GMFanuc Corp.

Lenguajes de programación de robots explícitos

Con la llegada de computadoras baratas y poderosas, la tendencia se ha orientado cada vez más hacia la programación de robots mediante programas escritos en lenguajes de programación de computadoras. Generalmente estos lenguajes tienen características especiales que se aplican a los problemas de la programación de manipuladores y por ello se les llama **lenguajes de programación de robots** (RPL). La mayoría de los sistemas que vienen equipados con un lenguaje de este tipo han retenido además una interfaz estilo dispositivo de enseñanza.

Dado que los lenguajes de programación de robots han tomado muchas formas, los dividiremos en tres categorías:

1. **Lenguajes de manipulación especializados**. Se han creado desarrollando un lenguaje completamente nuevo que, aunque se enfoca en áreas específicas de los robots, podría considerarse un lenguaje general de programación de computadoras. Un ejemplo es el lenguaje VAL desarrollado para controlar los robots industriales de Unimation, Inc. [1]. VAL se desarrolló específicamente como un lenguaje de control de manipuladores pues como lenguaje general de computadoras era bastante débil. Por ejemplo, no soportaba números de punto flotante o cadenas de caracteres, y las subrutinas no podían pasar argumentos. V-II, una versión más reciente, ya contaba con estas características [2]. V+, la encarnación actual de este lenguaje, incluye muchas características nuevas [13]. Otro ejemplo de un lenguaje de manipulación especializado es AL, desarrollado en la Universidad de Stanford [3]. Aunque el lenguaje AL es ahora una reliquia, de todas formas proporciona buenos ejemplos de algunas características que todavía no se encuentran en la mayoría de los lenguajes modernos (control de fuerza, paralelismo). Además, como fue construido en un entorno académico, hay referencias disponibles para describirlo [3]. Por estas razones seguimos haciendo referencia a este lenguaje.
2. **Biblioteca de robótica para un lenguaje de computación existente**. Estos lenguajes se han desarrollado a partir de un lenguaje popular de computación (por ejemplo, Pascal) al que se le agrega una biblioteca de subrutinas específicas para robots. El usuario escribe un programa en Pascal haciendo uso de llamadas frecuentes al paquete de subrutinas predefinido para las necesidades específicas del robot. Un ejemplo es AR-BASIC de American Cimflex [4], que esencialmente es una biblioteca de subrutinas para una implementación estándar de BASIC. JARS, desarrollado por el Laboratorio de Propulsión a Chorro de la NASA, es un ejemplo de dicho lenguaje de programación de robots basado en Pascal [5].
3. **Biblioteca de robótica para un nuevo lenguaje de propósito general**. Estos lenguajes de programación de robots se han desarrollado creando primero un nuevo lenguaje de propósito general como base de programación y después suministrando una biblioteca de subrutinas predefinidas, específicas para robots. Ejemplos de estos lenguajes de programación son RAPID, desarrollado por ABB Robotics [6]; AML, desarrollado por IBM [7], y KAREL, desarrollado por GMF Robotics [8].

Los estudios de programas de aplicación reales para células de trabajo robóticas han demostrado que un gran porcentaje de las instrucciones del lenguaje no son específicas para los robots [7], sino que la mayor parte de la programación de robots tiene que ver con la inicialización, prueba lógica y ramificación, comunicación, etcétera. Por

esta razón, podría desarrollarse una tendencia a alejarse del desarrollo de lenguajes especiales para la programación de robots y moverse hacia el desarrollo de extensiones para lenguajes generales, como en las categorías 2 y 3 antes mencionadas.

Lenguajes de programación a nivel de tarea

El tercer nivel de la metodología de programación de robots está relacionado con los **lenguajes de programación a nivel de tarea**. Estos lenguajes permiten al usuario ordenar directamente las submetas deseadas de la tarea, en vez de especificar los detalles de cada acción que el robot va a llevar a cabo. En dicho sistema el usuario puede incluir instrucciones en el programa de aplicación, con un nivel considerablemente más alto que en un lenguaje de programación de robots explícito. Un sistema de programación de robots a nivel de tarea debe tener la habilidad de realizar muchas tareas de planeación automáticamente. Por ejemplo, si se emite una instrucción para "sujetar el perno", el sistema debe planear una ruta que evite que el manipulador colisione con cualquier obstáculo circundante, elija automáticamente una buena posición de sujeción en el perno y sujete el perno. En contraste, en un lenguaje de programación de robots explícito el programador debe hacer todas estas elecciones.

Hay una buena distinción entre los lenguajes explícitos de programación de robots y los lenguajes de programación a nivel de tarea. Se están haciendo avances en los lenguajes explícitos de programación de robots para facilitar la programación, pero estas mejoras no pueden considerarse como componentes de un sistema de programación a nivel de tarea. Aún no existe la verdadera programación de manipuladores a nivel de tarea, pero desde hace tiempo es un tema de investigación [9,10] que continúa vigente.

12.3 UN EJEMPLO DE APLICACIÓN

La figura 12.2 muestra una célula de trabajo automatizada que completa un pequeño subensamblaje en un hipotético proceso de manufactura. La célula de trabajo consiste en un transportador bajo el control de una computadora que entrega una pieza de trabajo; una cámara conectada a un sistema de visión que se utiliza para localizar la pieza de trabajo en el transportador; un robot industrial (se muestra un PUMA 560) equipado con una muñeca detectora de fuerza; un pequeño alimentador ubicado en la superficie de trabajo que suministra otra pieza al manipulador; una prensa controlada por computadora que puede ser cargada y descargada por el robot; y una tarima sobre la que el robot coloca los ensamblajes terminados.

Todo el proceso está regulado por el controlador del manipulador en una secuencia, como se muestra a continuación:

1. Se indica al transportador que arranque y se detiene cuando el sistema de visión reporta que se ha detectado un soporte en el transportador.
2. El sistema de visión juzga la posición y la orientación del soporte en el transportador y lo inspecciona para verificar que no haya defectos, como un número incorrecto de hoyos.
3. Utilizando la salida del sistema de visión, el manipulador sujeta el soporte con una fuerza especificada y se verifica la distancia entre las puntas de los dedos para asegurar que el soporte se haya sujetado correctamente. En caso de algún error, el robot se hace a un lado y se repite la tarea de visión.
4. El soporte se coloca en el dispositivo fijo de la superficie de trabajo. En este punto se puede indicar al transportador que arranque de nuevo para colocar el siguiente soporte; esto es, los puntos 1 y 2 pueden empezar en paralelo con los siguientes pasos.

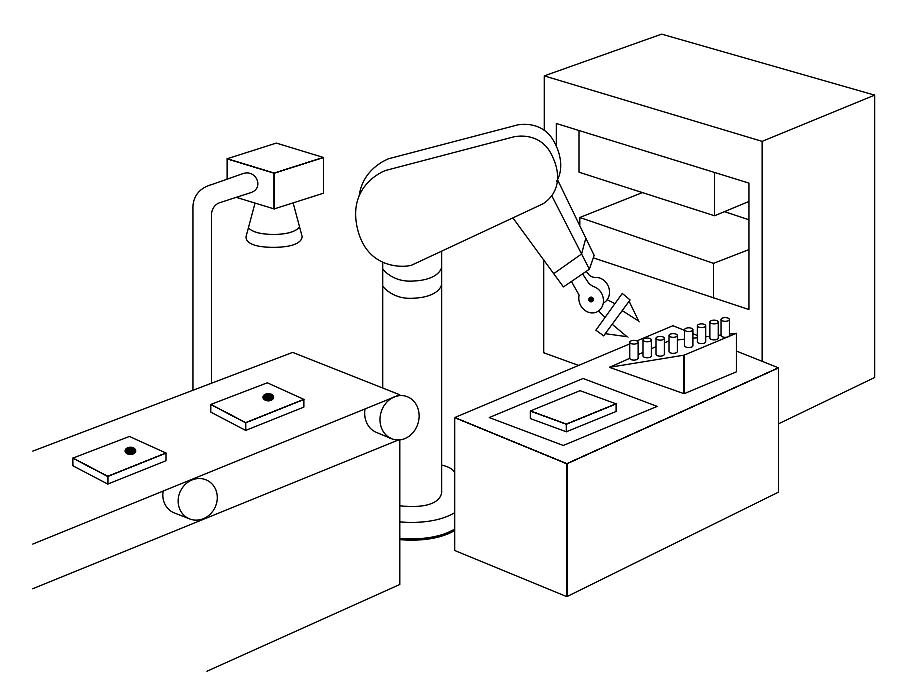


FIGURA 12.2: Una célula de trabajo automatizada que contiene un robot industrial.

5. Se recoge una chaveta del alimentador y se inserta parcialmente en un hoyo cónico en el soporte. Se utiliza el control de fuerza para realizar esta inserción y para realizar comprobaciones simples en cuanto a la terminación de esta tarea. (Si el alimentador de chavetas está vacío lo notifica a un operador y el manipulador espera hasta que el operador le ordene continuar).
6. El robot sujeta el conjunto soporte–chaveta y lo coloca en la prensa.
7. Se ordena a la prensa que actúe, y ésta presiona la chaveta completamente hacia dentro del soporte. La prensa indica que ha terminado y el soporte se coloca de nuevo en el dispositivo fijo para una inspección final.
8. El conjunto se revisa con un sensor de fuerza para verificar que la chaveta esté insertada correctamente. El manipulador detecta la fuerza de reacción cuando presiona hacia los lados en la chaveta y puede hacer las comprobaciones necesarias para saber qué tan insertada está la chaveta en el soporte.
9. Si el conjunto se encuentra en la posición correcta, el robot coloca la pieza terminada en la siguiente ubicación disponible en la tarima. Si está llena, se notifica al operador. Si el conjunto está mal ensamblado, se tira en el bote de basura.
10. Una vez terminado el paso 2 (iniciado anteriormente en paralelo) se vuelve al paso 3.

Éste es un ejemplo de una tarea que es posible para los robots industriales de la actualidad. Debe quedar claro que tal vez no sea posible definir dicho proceso mediante las técnicas de "enseñar mostrando". Por ejemplo, al tratar con tarimas es laborioso tener que programar todas las ubicaciones de todos los compartimentos; es preferible enseñar sólo la ubicación de la esquina y luego calcular las otras a partir de las dimensiones de la tarima. Además, por lo general es imposible especificar el señalamiento de interprocesos y establecer el paralelismo mediante el uso de un dispositivo de enseñanza

típico o de una interfaz estilo menú. Este tipo de aplicación necesita un enfoque basado en el uso de un lenguaje de programación de robots para procesar la descripción. (Vea el ejercicio 12.5). Por otro lado, esta aplicación es demasiado compleja como para que cualquier lenguaje existente a nivel de tarea trate con ella directamente. Es común que muchas excelentes aplicaciones se manejen con un método explícito de programación de robots. Tendremos el mismo ejemplo de aplicación en mente a medida que tratemos las características de los lenguajes de programación de robots.

12.4 REQUERIMIENTOS DE UN LENGUAJE DE PROGRAMACIÓN DE ROBOTS

Modelado del entorno

Por definición, los programas de manipulación deben involucrar el movimiento de objetos en espacio tridimensional, por lo que está claro que cualquier lenguaje de programación de robots necesita un medio para describir dichas acciones. El elemento más común de los lenguajes de programación de robots es la existencia de **tipos geométricos** especiales. Por ejemplo, se introducen los *tipos* para representar conjuntos de ángulos de articulación, posiciones cartesianas, orientaciones y tramas. A menudo hay operadores predefinidos disponibles que pueden manipular estos tipos. Las "tramas estándar" presentadas en el capítulo 3 podrían servir como un posible modelo del entorno: todos los movimientos se describen como una trama herramienta relativa a una trama estación, en donde las tramas de destino se crean a partir de expresiones arbitrarias que implican tipos geométricos.

Dado un entorno de programación de robots que soporta tipos geométricos, el robot y las demás máquinas, piezas y dispositivos fijos pueden modelarse si definimos variables con nombre asociadas con cada objeto. La figura 12.3 muestra parte de nuestro ejemplo de célula de trabajo con tramas unidas en ubicaciones relevantes a la tarea. Cada una de estas tramas se representaría con una variable de tipo "trama" en el programa del robot.

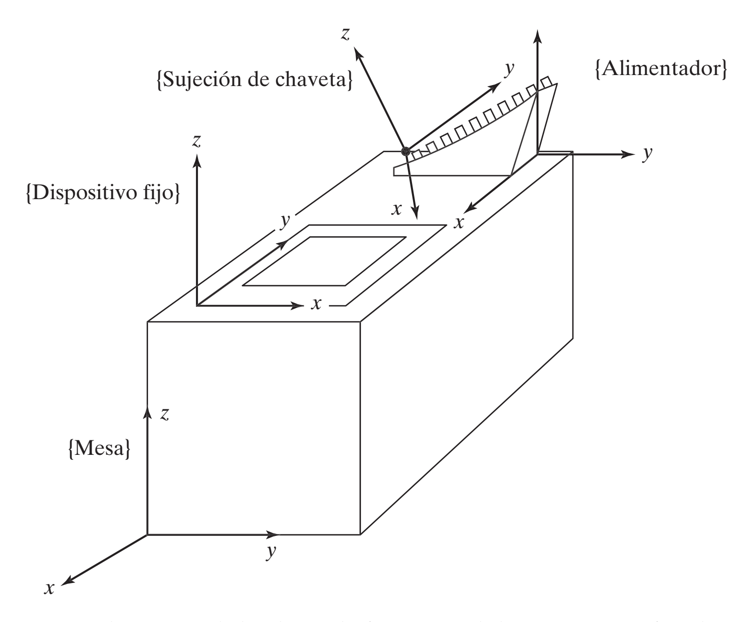


FIGURA 12.3: A menudo una célula de trabajo se modela en forma simple, como un conjunto de tramas unidas a objetos relevantes.

En muchos lenguajes de programación de robots, la habilidad para definir variables con nombre de distintos tipos geométricos y hacer referencia a ellas en el programa forma la base del modelo del entorno. Observe que las formas típicas de los objetos no son parte de dicho modelo del entorno, y tampoco los son las superficies, volúmenes, masas u otras propiedades. La extensión en la que se modelan los objetos en el entorno es una de las decisiones básicas al diseñar un sistema de programación de robots. La mayoría de los sistemas actuales soporta sólo el estilo que acabamos de describir.

Ciertos sistemas de modelado de entornos permiten la noción de **fijaciones** entre objetos con nombre [3]; esto es, el sistema puede recibir una notificación de que se han "fijado" dos o más objetos con nombre; a partir de ese momento, si un objeto se mueve explícitamente con una instrucción del lenguaje, se moverán todos los objetos fijados a éste. Así, una vez que se ha insertado la chaveta en el agujero del soporte en nuestra aplicación, el sistema recibirá una notificación (mediante una instrucción del lenguaje) de que se han fijado estos dos objetos. Los movimientos subsiguientes del soporte (es decir, los cambios en el valor de la variable de trama "soporte") harán que el valor almacenado para la variable "chaveta" se actualice al mismo tiempo.

Idealmente, un sistema de modelado de entornos incluiría mucha más información sobre los objetos con los que tiene que tratar el manipulador y sobre el manipulador en sí. Por ejemplo, considere un sistema en el que los objetos se describen mediante modelos estilo CAD que representan la forma espacial de cualquier objeto mediante las definiciones de sus bordes, sus superficies o su volumen. Con estos datos disponibles para el sistema, es más factible implementar muchas de las características de un sistema de programación a nivel de tarea. En el capítulo 13 hablaremos más sobre estas posibilidades.

Especificación del movimiento

Una función muy básica de un lenguaje de programación de robots es permitir la descripción de los movimientos que se espera que realice. El usuario interactúa con los planeadores de rutas y los generadores de estilo descritos en el capítulo 7 mediante instrucciones de movimiento que permiten especificar puntos vía, el punto de destino y si se va a utilizar el movimiento interpolado de las articulaciones o en línea recta cartesiana. Además, el usuario podría tener el control sobre la velocidad o duración de un movimiento.

Para ilustrar varias sintaxis para las primitivas del movimiento, consideraremos los siguientes movimientos de un manipulador como ejemplo: (1) moverse a la posición "destino1"; después (2) moverse en línea recta hasta la posición "destino2"; después (3) moverse, sin detenerse, a través de "vía1" y detenerse completamente en "destino3". Suponiendo que todos esos puntos de ruta ya se han enseñado o se han descrito textualmente, este segmento de programa se escribiría así:

En VAL II,

```
move destino1
move destino2
move via1
move destino3
```

En AL (aquí se controla el manipulador "garm"),

```
move garm to destino1;
move garm to destino2 linearly;
move garm to destino3 via via1;
```

La mayoría de los lenguajes tiene una sintaxis similar para instrucciones de movimiento simples como éstas. Las diferencias en las primitivas de movimiento básico de un lenguaje de programación de robots a otro se hacen más aparentes si consideramos características como las siguientes:

1. La habilidad para realizar matemáticas con tipos estructurados tales como tramas, vectores y matrices de rotación.
2. La habilidad de describir entidades geométricas como tramas en varias representaciones convenientes; junto con la habilidad de realizar conversiones entre las representaciones.
3. La habilidad de proporcionar restricciones sobre la duración o la velocidad de un movimiento específico; por ejemplo, muchos sistemas permiten al usuario establecer la velocidad a una fracción del máximo, pero pocas le permiten especificar directamente una duración deseada o una velocidad de articulación máxima.
4. La habilidad de especificar destinos en relación con varias tramas, incluyendo las tramas definidas por el usuario y las tramas en movimiento (en un transportador, por ejemplo).

Flujo de ejecución

Al igual que en los lenguajes convencionales de programación de computadoras, un sistema de programación de robots permite al usuario especificar el flujo de la ejecución; esto es, los conceptos tales como prueba y ramificación, ciclos, llamadas a subrutinas e incluso interrupciones se encuentran generalmente en los lenguajes de programación de robots.

Más que en la mayoría de las aplicaciones computacionales, el procesamiento en paralelo es generalmente importante en aplicaciones de células de trabajo automatizadas. En primer lugar, es muy común que dos o más robots se utilicen en una sola célula de trabajo y que trabajen de manera simultánea para reducir el tiempo de ciclo del proceso. Incluso en las aplicaciones de un solo robot, como la que se muestra en la figura 12.2, el controlador del robot debe controlar otro equipo de la célula de trabajo en forma paralela. Por ende, a menudo se encuentran las primitivas *señal* y *espera* en los lenguajes de programación de robots, y ocasionalmente se proporcionan construcciones de ejecución en paralelo más sofisticadas [3].

Otro hecho frecuente es la necesidad de monitorear varios procesos con algún tipo de sensor. Después, ya sea mediante la interrupción o el sondeo, el sistema de robot debe ser capaz de responder a ciertos eventos detectados por los sensores. La habilidad de especificar dichos **monitores de eventos** es soportada por algunos lenguajes de programación de robots [2, 3].

Entorno de programación

Al igual que con cualquier lenguaje computacional, un buen entorno de programación fomenta la productividad del programador. La programación de manipuladores es difícil y tiende a ser muy interactiva, con muchas sesiones de prueba y error. Si se obligara al usuario a repetir continuamente el ciclo de "editar-compilar-ejecutar" de los lenguajes compilados, la productividad sería baja. Por lo tanto, la mayoría de los lenguajes de programación ahora son *interpretados*, de manera que las instrucciones individuales puedan ejecutarse de una en una durante el desarrollo y la depuración del programa. Muchas de las instrucciones de un lenguaje ocasionan el movimiento de un dispositivo físico, por lo que la pequeña cantidad de tiempo requerido para interpretar las instrucciones del lenguaje es insignificante. También se requiere el soporte típico de programación: editores de texto, depuradores y un sistema de archivos.

Integración de sensores

Una parte extremadamente importante de la programación de robots tiene que ver con la interacción con sensores. El sistema debe tener, como mínimo, la capacidad de consultar sensores de tacto y fuerza, y de utilizar la respuesta en instrucciones "if-then-else" (si-entonces). La habilidad de especificar monitores de eventos para vigilar las transiciones en dichos sensores en un modo de *segundo plano* es también muy útil.

La integración con un sistema de visión permite que el sistema de visión envíe al sistema manipulador las coordenadas de un objeto de interés. Por ejemplo, en nuestra aplicación de ejemplo, un sistema de visión ubica los soportes en la banda transportadora y regresa al controlador del manipulador su posición y orientación relativas a la cámara. Se conoce la trama de la cámara relativa a la trama estación, por lo que puede calcularse una trama de destino deseada para el manipulador a partir de esta información.

Algunos sensores podrían ser parte de otro equipo en la célula de trabajo; por ejemplo, algunos controladores de robot pueden utilizar la entrada proveniente de un sensor unido a una banda transportadora, de manera que el manipulador pueda rastrear el movimiento de la banda y adquirir objetos de ésta a medida que se va moviendo [2].

La interfaz para las capacidades de control de fuerza, como se vio en el capítulo 9, se da a través de instrucciones especiales del lenguaje que permiten al usuario especificar estrategias de fuerza [3]. Dichas estrategias son, por necesidad, una parte integrada del sistema de control del manipulador; el lenguaje de programación de robots simplemente sirve como interfaz para esas capacidades. Para programar robots que hagan uso del control de fuerza activo se podrían requerir otras características especiales, tales como la habilidad de mostrar datos de fuerza recolectados durante un movimiento restringido [3].

En sistemas que soportan el control de fuerza activo, la descripción de la aplicación de fuerza deseada podría convertirse en parte de la especificación del movimiento. El lenguaje AL describe el control de fuerza activo en las primitivas de movimiento, especificando seis componentes de rigidez (tres de traslación y tres de rotación) y una fuerza de desvío. De esta manera, la rigidez aparente del manipulador es programable. Para aplicar una fuerza, por lo general la rigidez se hace cero en esa dirección y se especifica una fuerza de desvío; por ejemplo,

```
move garm to destino
with stiffness=(80, 80, 0, 100, 100, 100)
with force=20*ounces along zhat;
```

12.5 PROBLEMAS PECULIARES PARA LOS LENGUAJES DE PROGRAMACIÓN DE ROBOTS

Los avances en años recientes han ayudado, pero aún es difícil programar robots. La programación de robots comparte todos los problemas de la programación convencional de computadoras, más ciertas dificultades adicionales ocasionadas por los efectos del mundo físico [12].

Comparación entre el modelo del entorno y la realidad externa

Una característica central de un sistema de programación de robots es el modelo del entorno que se mantiene internamente en la computadora. Aun y cuando este modelo es considerablemente simple, hay grandes dificultades para asegurar que concuerde con la realidad física que trata de modelar. Las discrepancias entre el modelo interno y la realidad externa hacen que los objetos se sujeten mal o no se sujeten, produzcan colisiones y otros problemas más sutiles.

Esta correspondencia entre el modelo interno y el externo debe establecerse para el estado inicial del programa, y debe mantenerse durante su ejecución. Durante la programación inicial o la depuración es generalmente responsabilidad del usuario asegurarse que el estado representado en el programa corresponda con el estado físico de la célula de trabajo. A diferencia de la programación más convencional, en donde sólo necesitan guardarse variables internas y restaurarse para restablecer una situación previa, en la programación de robots los objetos físicos comúnmente deben reposicionarse.

Además de la incertidumbre inherente en la posición de cada objeto, el manipulador en sí está limitado a un cierto grado de precisión. Muy a menudo los pasos en el ensamblaje requieren que el manipulador haga movimientos que requieren mayor precisión de la que es capaz; un ejemplo común es insertar una chaveta en un hoyo en el que el espacio libre es de una magnitud menor que la precisión posicional del manipulador. Para complicar más las cosas, la precisión del manipulador generalmente varía sobre su espacio de trabajo.

Al tratar con esos objetos cuyas ubicaciones no se conocen exactamente, es esencial refinar de alguna forma la información posicional. Esto puede hacerse algunas veces con sensores (por ejemplo, de visión, de tacto) o utilizando estrategias de fuerza apropiadas para los movimientos restringidos.

Durante la depuración de los programas del manipulador, es muy útil poder modificar el programa y después respaldar y probar un procedimiento otra vez. Para respaldar hay que restaurar el manipulador y los objetos que se están manipulando a un estado anterior. No obstante, al trabajar con objetos físicos no siempre es fácil, o incluso posible, deshacer una acción. Algunos ejemplos son las operaciones de pintura, remachado, perforación o soldadura, que producen una modificación física de los objetos que se están manipulando. Por lo tanto, podría requerirse que el usuario obtenga una nueva copia del objeto para reemplazar el anterior que se ha modificado. Además, es probable que algunas de las operaciones realizadas justo antes de la que se va a reintentar también necesiten repetirse para establecer el estado apropiado requerido antes de que pueda reintentarse con éxito la operación deseada.

Sensibilidad al contexto

La **programación ascendente (*bottom-up programming*)** es un método estándar para escribir un programa computacional extenso en el que se desarrollan pequeñas piezas de bajo nivel de un programa y luego se juntan en piezas más grandes, formando un programa completo. Para que este método funcione es esencial que las piezas pequeñas sean relativamente insensibles a las instrucciones del lenguaje que las precede y que no haya suposiciones con relación al contexto en el que se ejecutan. Para la programación de manipuladores éste no es comúnmente el caso; el código que trabajaba confiablemente al probarse aislado con frecuencia falla cuando se coloca dentro del contexto de un programa más grande. Estos problemas generalmente surgen de las dependencias en la configuración del manipulador y la velocidad de los movimientos.

Los programas de manipuladores pueden ser altamente sensibles a las condiciones iniciales; por ejemplo, la posición inicial del manipulador. En trayectorias de movimiento la posición inicial podría también influenciar la velocidad con la que se moverá el brazo durante cierta parte crítica del movimiento. Por ejemplo, estas aseveraciones se cumplen para los manipuladores que siguen las rutas del tipo trazador cúbico en el espacio de articulación que estudiamos en el capítulo 7. Estos efectos pueden algunas veces tratarse mediante un cuidado apropiado en la programación, pero a menudo dichos

problemas no surgen sino hasta después que se han depurado en forma aislada las declaraciones iniciales del lenguaje y luego se unen con las instrucciones que las preceden.

Debido a una precisión insuficiente del manipulador, es muy probable que un segmento de programa escrito para realizar una operación en cierta ubicación necesite ajustarse (es decir, volver a enseñar las posiciones, etcétera) para hacerlo que funcione en una ubicación distinta. Los cambios de ubicación dentro de la célula de trabajo producen cambios en la configuración del manipulador para llegar a las ubicaciones de destino. Dichos intentos de reubicar los movimientos del manipulador dentro de la célula de trabajo ponen a prueba la precisión de los comportamientos cinemáticos del manipulador y del sistema de servo, y con frecuencia surgen problemas. Dicha reubicación podría provocar un cambio en la configuración cinemática del manipulador; por ejemplo, del hombro izquierdo al hombro derecho, o del codo hacia arriba al codo hacia abajo. Lo que es más, estos cambios en la configuración podrían ocasionar grandes movimientos del brazo durante lo que había sido previamente un movimiento corto y simple.

Es muy probable que la naturaleza de la forma espacial de las trayectorias cambie a medida que se ubican las rutas en distintas porciones del espacio de trabajo del manipulador. Esto es especialmente cierto para los métodos de trayectorias en espacio de articulación, pero el uso de los esquemas de ruta cartesiana también puede producir problemas cuando hay singularidades cerca.

Al probar el movimiento de un manipulador por primera vez, a menudo es conveniente hacer que el manipulador se mueva lentamente. Esto da al usuario la oportunidad de detener el movimiento si parece estar a punto de provocar una colisión; también le permite inspeccionar el movimiento muy de cerca. Una vez que el movimiento ha pasado por cierta depuración inicial a una velocidad menor, es conveniente aumentar las velocidades del movimiento. Al hacer esto algunos aspectos del movimiento podrían cambiar. Las limitaciones en la mayoría de los sistemas de control de manipuladores producen errores de servo mayores, los cuales son de esperarse si se sigue la trayectoria más rápida. Además, en situaciones de control de fuerza que impliquen un contacto con el entorno, los cambios de velocidad pueden modificar completamente las estrategias de fuerza requeridas.

La configuración del manipulador también afecta la delicadeza y precisión de las fuerzas que pueden aplicarse con él. Ésta es una función de qué tan bien condicionado está el jacobiano del manipulador en cierta configuración, algo generalmente difícil de considerar al desarrollar programas para robots.

Recuperación de errores

Otra consecuencia directa de trabajar con el mundo físico es que los objetos podrían no encontrarse exactamente en donde deberían y, por lo tanto, los movimientos que se realizan con esos objetos podrían fallar. Parte de la programación de manipuladores implica el tratar de tomar esto en cuenta y hacer las operaciones de ensamblaje lo más robustas que sea posible pero, aún así, es probable que haya errores, y un aspecto importante de la programación de manipuladores es cómo recuperarse de estos errores.

Casi cualquier instrucción de movimiento en el programa del usuario puede fallar, por una gran variedad de razones. Algunas de las causas más comunes son que los objetos se desplacen o se resbalen de la mano, que un objeto no se encuentre en donde debería estar, que se atasque el objeto durante una inserción y que no se pueda localizar un hoyo.

El primer problema que surge para la recuperación de errores es identificar que efectivamente haya ocurrido un error. Como los robots generalmente tienen capacidades de detección y razonamiento bastante limitadas, a menudo la *detección de errores* es

difícil. Para poder detectar un error, un programa de robot debe contener cierto tipo de prueba explícita. Esta prueba podría implicar el comprobar la posición del manipulador para ver que se encuentre en el rango apropiado; por ejemplo, cuando se realiza una inserción, la falta de cambio de posición podría indicar un atascamiento, o demasiado cambio podría indicar que no se pudo localizar el hoyo o que el objeto se resbaló de la mano. Si el sistema de manipulador cuenta con cierto tipo de capacidades visuales, entonces podría tomar una foto y revisar la presencia o ausencia de un objeto y, si el objeto está presente, reportar su ubicación. Otras comprobaciones podrían implicar el uso de fuerza, como pesar la carga que se lleva para revisar que el objeto aún esté allí y no se haya caído, o revisar que una fuerza de contacto permanezca dentro de ciertos límites durante un movimiento.

Todas las instrucciones de movimiento en el programa podrían fallar, por lo que estas comprobaciones explícitas pueden ser bastante complicadas y pueden ocupar más espacio que el resto del programa. Es en extremo difícil tratar de lidiar con todos los posibles errores; generalmente sólo se comprueban las instrucciones con más probabilidad de fallar; y este proceso requiere de cierta cantidad de interacción y prueba parcial con el robot durante la etapa de desarrollo del programa.

Una vez que se ha detectado un error, puede hacerse un intento por recuperarse del mismo; el manipulador puede hacerlo bajo el control del programa o tal vez el usuario tenga que intervenir, o podría requerirse cierta combinación de las dos cosas. En cualquier caso el intento de recuperación podría producir nuevos errores. Es fácil ver cómo el código para recuperarse de los errores puede convertirse en la parte principal del programa del manipulador.

El uso del paralelismo en los programas de manipuladores puede complicar aún más la recuperación después de errores. Cuando varios procesos se ejecutan concurrentemente y uno de ellos provoca un error, podría afectar otros procesos. En muchos casos es posible hacer retroceder el proceso que provoca el error, permitiendo a los otros continuar. Otras veces es necesario reiniciar varios o todos los procesos en ejecución.

BIBLIOGRAFÍA

[1] B. Shimano, "VAL: A Versatile Robot Programming and Control System", *Proceedings of COMPSAC 1979*, Chicago, noviembre de 1979.

[2] B. Shimano, C. Geschke y C. Spalding, "VAL II: A Robot Programming Language and Control System", *SME Robots VIII Conference*, Detroit, junio de 1984.

[3] S. Mujtaba y R. Goldman, *AL User´s Manual*, 3a. edición, Stanford Department of Computer Science, Reporte No. STAN-CS-81-889, diciembre de 1981.

[4] A. Gilbert y otros, *AR-BASIC: An Advanced and User Friendly Programming System for Robots*, American Robot Corporation, junio de 1984.

[5] J. Craig, "JARS—JPL Autonomous Robot System: Documentation and Users Guide", JPL memorándum entre oficinas, septiembre de 1980.

[6] ABB Robotics, "The RAPID Language", en *SC4Plus Controller Manual*, ABB Robotics, 2002.

[7] R. Taylor, P. Summers y J. Meyer, "AML: A Manufacturing Language", *International Journal of Robotics Research*, Vol. 1, núm. 3, otoño de 1982.

[8] FANUC Robotics, Inc., "KAREL Language Reference", FANUC Robotics North America, Inc, 2002.

[9] R. Taylor, "A Synthesis of Manipulator Control Programs from Task-Level Specifications", Stanford University AI Memo 282, julio de 1976.

[10] J. C. LaTombe, "Motion Planning with Uncertainty: On the Preimage Backchaining Approach", en *The Robotics Review*, O. Khatib, J. Craig y T. Lozano-Perez, editores, MIT Press, Cambridge, MA, 1989.

[11] W. Gruver and B. Soroka, "Programming, High Level Languages", en *The International Encyclopedia of Robotics*, R. Dorf y S. Nof, Editores, Wiley Interscience, Nueva York, 1988.

[12] R. Goldman, *Design of an Interactive Manipulator Programming Environment*, UMI Research Press, Ann Arbor, MI, 1985.

[13] Adept Technology, *V+ Language Reference*, Adept Technology, Livermore, CA, 2002.

EJERCICIOS

12.1 [15] Escriba un programa de robot (en el lenguaje de su elección) para recoger un bloque de la ubicación A y colocarlo en la ubicación B.

12.2 [20] Con simples comandos en español, que podrían formar la base de un programa de robot, describa cómo atar las agujetas de sus zapatos.

12.3 [32] Diseñe la sintaxis de un nuevo lenguaje de programación de robots. Incluya formas de proporcionar la duración o las velocidades de las trayectorias de movimiento, haga instrucciones de E/S para los periféricos, proporcione comandos para controlar la pinza de sujeción y produzca comandos de detección de fuerza (es decir, de movimiento protegido). Puede omitir el control de fuerza y el paralelismo.

12.4 [28] Extienda la especificación del nuevo lenguaje de programación de robots que inició en el ejercicio 12.3, agregando la sintaxis para el control de fuerza y para el paralelismo.

12.5 [38] Escriba un programa, en un lenguaje de programación de robots que esté comercialmente disponible, para realizar la aplicación descrita en la sección 12.3. Haga todas las suposiciones razonables en relación con las conexiones de E/S y otros detalles.

12.6 [28] Utilizando cualquier lenguaje de robots, escriba una rutina general para descargar una tarima de tamaño arbitrario. La rutina deberá mantener un registro de indización a través de la tarima y deberá indicar a un operador humano cuando ésta se encuentre vacía. Suponga que las piezas se descargan en una banda transportadora.

12.7 [35] Usando cualquier lenguaje de robot, escriba una rutina general para descargar una tarima fuente de tamaño arbitrario, y para cargar una tarima destino de tamaño arbitrario. La rutina deberá llevar el registro de indización a través de las tarimas y deberá indicar a un operador humano cuando la tarima fuente esté vacía y cuando la tarima destino esté llena.

12.8 [35] Usando cualquier lenguaje de programación de robots que sea capaz de realizarlo, escriba un programa que emplee el control de fuerza para llenar una cajetilla con 20 cigarrillos. Suponga que el manipulador tiene una precisión de aproximadamente 0.25 pulgadas, por lo que deberá usarse el control de fuerza para muchas operaciones. Los cigarrillos se presentan en una banda transportadora, y un sistema de visión devuelve sus coordenadas.

12.9 [35] Usando cualquier lenguaje de programación de robots capaz de llevarlo a cabo, escriba un programa para ensamblar la porción portátil de un teléfono estándar. Los seis componentes (auricular, micrófono, bocina, dos tapas y un cable) llegan en un *kit*, esto es, una tarima especial que sostiene una pieza de cada componente. Suponga que hay un dispositivo fijo en el que puede colocarse el auricular para que lo sostenga. Haga todas las demás suposiciones necesarias.

12.10 [33] Escriba un programa de robot que utilice dos manipuladores. Uno, llamado GARM, tiene un efector final especial diseñado para sostener una botella de vino. El otro brazo, llamado BARM, sostendrá una copa de vino y está equipado con una muñeca detectora de fuerza que puede usarse para indicar a GARM que deje de verter cuando detecte que la copa está llena.

EJERCICIO DE PROGRAMACIÓN (PARTE 12)

Cree una interfaz de usuario para los demás programas que ha desarrollado, escribiendo unas cuantas subrutinas en Pascal. Una vez definidas estas subrutinas, un "usuario" podría escribir un programa en Pascal que contenga llamadas a estas subrutinas, para realizar la simulación de una aplicación de robot en 2-D.

Defina primitivas que permitan al usuario ajustar las tramas estación y herramienta; a saber,

```
ajustarestacion(SrelB:vec3);

ajustarherramienta(TrelW:vec3);
```

en donde "SrelB" indica la trama estación relativa a la trama base del robot y "TrelW" define la trama herramienta relativa a la trama muñeca del manipulador. Defina las primitivas de movimiento

```
moveto(destino:vec3);

moveto(incremento:vec3);
```

donde "destino" es una especificación de la trama de destino relativa a la trama estación, e "incremento" es una especificación de la trama de destino relativa a la trama herramienta actual. Permita la descripción de rutas multisegmento cuando el usuario llame primero a la función "pathmode", luego especifique movimientos a los puntos vía, y finalmente diga "runpath"; por ejemplo,

```
pathmode; (* entrar en el modo ruta *)

moveto(destino1);

moveto(destino2);

runpath; (* ejecutar la ruta sin detenerse en el destino1 *)
```

Escriba un simple programa de "aplicación" y haga que su sistema imprima la ubicación del brazo cada *n* segundos.

CAPÍTULO 13

Sistemas de programación fuera de línea

13.1 INTRODUCCIÓN

Definiremos un sistema de **programación fuera de línea** (OLP, por sus siglas en inglés) como un lenguaje de programación que se ha extendido lo suficiente, generalmente mediante gráficos computacionales, como para que el desarrollo de programas de robot pueda llevarse a cabo sin necesidad de tener acceso al mismo robot.[1] Los sistemas de programación fuera de línea son importantes como ayuda en la programación de la automatización industrial contemporánea, y también como plataformas de investigación robótica. Hay que considerar muchas cuestiones en el diseño de dichos sistemas. En este capítulo presentaremos primero una discusión sobre estas cuestiones [1] y después veremos más de cerca un sistema de este tipo [2].

Durante los últimos 20 años, el crecimiento del mercado de robots industriales no ha sido tan rápido como una vez se pronosticó; una de las principales razones es que los robots aún son muy difíciles de utilizar. Se requiere de gran cantidad de tiempo y experiencia para instalar un robot en una aplicación específica y hacer que el sistema esté listo para la producción. Por varias razones, este problema es más severo en algunas aplicaciones que en otras; por lo tanto, vemos que ciertas áreas (por ejemplo, la soldadura por punto y la pintura con pistola de aire) se están automatizando con robots mucho más pronto que otros dominios de aplicación (por ejemplo, el ensamblaje). Parece que la falta de implementadores de sistemas de robots con suficiente capacitación está limitando el crecimiento en algunas áreas de aplicación (si no es que en todas). En ciertas compañías de fabricación, la gerencia alienta el uso de robots a un nivel mayor que el que pueden realizar los ingenieros de aplicaciones. Además, un gran porcentaje de los robots entregados se están usando en formas que no aprovechan completamente sus capacidades. Estos síntomas indican que los robots industriales de la actualidad no son lo suficientemente fáciles de usar como para permitir que se instalen y se programen con éxito en forma oportuna.

Muchos factores hacen que la programación de robots sea una tarea difícil. En primer lugar está relacionada intrínsecamente con la programación de computadoras en general y, por lo tanto, comparte muchos de los problemas de ese campo; pero la programación de robots, o de cualquier máquina programable, tiene problemas específicos

[1] El capítulo 13 es una versión editada de dos artículos: uno reimpreso con permiso de *International Symposium of Robotics Research*, R. Bolles y B. Roth (editores), 1988 (ref [1]); el otro de *Robotics: The Algorithmic Perspective*, P. Agarwal y otros (editores), 1998 (ref [2]).

que hacen que el desarrollo de software listo para la producción sea aún más difícil. Como vimos en el capítulo anterior, la mayor parte de estos problemas surgen del hecho de que un manipulador de robot interactúa con su entorno físico [3]. Incluso hasta los sistemas de programación simples mantienen un “modelo global” de este entorno físico en la forma de ubicaciones de objetos, y tienen “conocimiento” de la presencia y ausencia de varios objetos codificados en las estrategias del programa. Durante el desarrollo de un programa de robot (y especialmente después, durante su uso en la producción), es necesario que el sistema de programación mantenga el modelo interno en correspondencia con el estado actual del entorno del robot. La depuración interactiva de programas con un manipulador requiere reiniciar frecuentemente el estado del entorno del robot; las piezas, las herramientas y demás objetos necesitan regresarse a sus posiciones iniciales. Este proceso de reiniciar el estado se vuelve especialmente difícil (y algunas veces costoso) cuando el robot realiza una operación irreversible en una o más piezas (por ejemplo, perforación o enrutamiento). El efecto más espectacular de la presencia del entorno físico es cuando un error del programa se manifiesta en cierta operación irreversible no intencional sobre las piezas, las herramientas o incluso sobre el mismo manipulador.

Aunque existen dificultades para mantener un modelo interno preciso del entorno del manipulador, parece que no hay duda de que se generan grandes beneficios. Áreas completas de investigación de sensores (tal vez la más notable sea la visión por computadora) se enfocan en desarrollar técnicas mediante las cuales se puedan verificar, corregir o descubrir los modelos del entorno. Evidentemente, para poder aplicar cualquier algoritmo computacional al problema de generación de comandos para robot, el algoritmo necesita acceso a un modelo del robot y de su entorno.

En el desarrollo de sistemas de programación para robots, los avances en el poder de las técnicas de programación parecen estar directamente vinculados con la sofisticación del modelo interno al que hace referencia el lenguaje de programación. Los primeros sistemas de robots operando en el espacio de articulación, del tipo “enseñar mostrando”, empleaban un modelo limitado del mundo, por lo que había maneras muy limitadas en las que el sistema podía ayudar al programador a lograr una tarea. Los controladores de robot ligeramente más sofisticados incluían modelos cinemáticos, de manera que el sistema podía cuando menos ayudar al usuario a mover las articulaciones para lograr movimientos cartesianos. Los lenguajes de programación de robots (RPL) evolucionaron para soportar muchos tipos de datos y operaciones distintos, que el programador puede utiliza según necesite, para modelar los atributos del entorno y calcular acciones para el robot. Algunos RPL soportan primitivas de modelado del entorno tales como fijaciones, tipos de datos para fuerzas y momentos, y otras características [4].

Los actuales lenguajes de programación de robots podrían llamarse “lenguajes de programación explícitos”, ya que el ingeniero de aplicación debe programar cada acción que realiza el sistema. Del otro lado del espectro están los llamados sistemas de programación a nivel de tarea (TLP), en los que el programador puede declarar objetivos de alto nivel tales como “insertar el perno” o tal vez hasta “fabricar el horno tostador”. Estos sistemas utilizan técnicas de la investigación en inteligencia artificial para generar planes de movimiento y estrategia de manera automática. No obstante, los lenguajes a nivel de tarea así de sofisticados no existen todavía; varias piezas de dichos sistemas se encuentran actualmente en desarrollo por parte de los investigadores [5]. Los sistemas de programación a nivel de tarea requieren un modelo muy completo del robot y de su entorno para realizar las operaciones de planeación automatizadas.

Aunque este capítulo se enfoca hasta cierto punto en el problema específico de la programación de robots, la noción de un sistema OLP se extiende a cualquier dispositivo

programable en la fábrica. Un argumento que surge comúnmente en favor es que un sistema OLP no acapara el equipo de producción cuando necesita reprogramarse; por ende, las fábricas automatizadas pueden permanecer en modo de producción un mayor porcentaje del tiempo. También sirven como un vehículo natural para enlazar las bases de datos de diseño asistido por computadora (CAD) que se utilizan en la fase de diseño del desarrollo de un producto con la fabricación del mismo. En ciertas aplicaciones, el uso directo de datos de diseño CAD puede reducir considerablemente el tiempo de programación requerido para la maquinaria de fabricación.

La programación de robots fuera de línea ofrece otros beneficios potenciales, de los cuales algunos ya están siendo apreciados por los usuarios de robots industriales. Hemos hablado sobre algunos de los problemas asociados con la programación de robots, y la mayoría tienen que ver con el hecho de que una célula de trabajo física externa está siendo manipulada por el programa del robot. Esto hace que el proceso de retroceder para probar distintas estrategias sea tedioso. La programación de robots en simulación ofrece una manera de mantener todo lo relacionado con el trabajo de programación de una manera estrictamente interna en una computadora hasta que la aplicación esté casi completa. Con este enfoque, muchos de los problemas peculiares de la programación de robots tienden a desaparecer.

Los sistemas de programación fuera de línea deberían servir como ruta de crecimiento natural de los sistemas de programación explícitos a los sistemas de programación a nivel de tarea. El sistema OLP más simple es sólo una extensión gráfica para un lenguaje de programación de robots, pero de ahí puede extenderse hacia un sistema de programación a nivel de tarea. Esta extensión gradual se logra proporcionando soluciones automatizadas a varias subtareas (a medida que estas soluciones estén disponibles) y dejando que el usuario las utilice para explorar opciones en el entorno simulado. Hasta que descubramos cómo construir sistemas a nivel de tarea, el usuario deberá permanecer en el ciclo para evaluar subtareas planeadas automáticamente y guiar el desarrollo del programa de aplicación. Si tomamos este enfoque, un sistema OLP sirve como una base importante para la investigación y el desarrollo de sistemas de planeación a nivel de tarea y, evidentemente, para apoyar su trabajo muchos investigadores han desarrollado varios componentes de un sistema OLP (por ejemplo, modelos en 3-D y pantallas gráficas, posprocesadores de lenguaje). En consecuencia, los sistemas OLP deberían ser una herramienta útil en la investigación, así como una ayuda en la práctica industrial actual.

13.2 CUESTIONES BÁSICAS DE LOS SISTEMAS OLP

Esta sección trata sobre muchos de los puntos que deben considerarse en el diseño de un sistema OLP. La colección de temas descritos le ayudará a establecer el alcance de la definición de un sistema OLP.

Interfaz de usuario

Una motivación importante para desarrollar un sistema OLP es crear un entorno que facilite la programación de manipuladores, por lo que la interfaz de usuario es de crucial importancia. Sin embargo, otra de las principales motivaciones es eliminar la dependencia en el uso del equipo físico durante la programación. En una consideración inicial, estas dos metas parecen estar en conflicto; los robots ya son bastante difíciles de programar cuando se les tiene enfrente, así que ¿cómo puede ser más fácil sin la presencia del dispositivo físico? Esta pregunta toca la esencia del problema de diseño de sistemas OLP.

Los fabricantes de robots industriales han aprendido que un gran porcentaje del personal de manufactura no puede utilizar correctamente los RPL que proveen con sus robots. Por ésta y otras razones históricas, muchos robots industriales vienen equipados con una interfaz de dos niveles [6], una para programadores y otra para no programadores. Los no programadores utilizan un dispositivo de enseñanza e interactúan directamente con el robot para desarrollar los programas. Los programadores escriben código en el RPL e interactúan con el robot para poder enseñarle los puntos de trabajo y depurar el flujo del programa. En general, estos dos enfoques para el desarrollo de programas intercambian la facilidad de uso por la flexibilidad.

Cuando se ve como una extensión de un RPL, un sistema OLP contiene por naturaleza un RPL como subconjunto de su interfaz de usuario. Este RPL debe proveer características que se consideren valiosas en los sistemas de programación de robots. Por ejemplo, para usarse como un RPL, los **lenguajes interactivos** son mucho más productivos que los compilados, los cuales obligan al usuario a pasar por el ciclo de "editar-compilar-ejecutar" en cada modificación del programa.

La porción de la interfaz de usuario correspondiente al lenguaje hereda mucho de los RPL "tradicionales"; es la interfaz de nivel más bajo (es decir, más fácil de usar) que debe considerarse cuidadosamente en un sistema OLP. Un componente central de esta interfaz es una vista de gráficos computacionales del robot que se está programando y de su entorno. Utilizando un dispositivo señalador tal como un **ratón**, el usuario puede indicar varias ubicaciones u objetos en la pantalla de gráficos. El diseño de la interfaz de usuario controla exactamente la forma en que el usuario interactúa con la pantalla para especificar un programa de robot. El mismo dispositivo señalador puede indicar elementos en un "menú" para poder invocar varias funciones.

Una primitiva central es la de enseñar a un robot un punto de trabajo o "trama" que tenga seis grados de libertad por medio de la interacción con la pantalla de gráficos. La disponibilidad de modelos 3-D de los dispositivos fijos y las piezas de trabajo en el sistema OLP facilita esta tarea considerablemente. La interfaz provee al usuario con los medios para indicar las ubicaciones en las superficies, permitiendo que la orientación de la trama tome una normal a la superficie local, y luego proporciona métodos para desplazar, reorientar y así sucesivamente. Dependiendo de los detalles específicos de la aplicación, dichas tareas se especifican muy fácilmente mediante la ventana de gráficos en el mundo simulado.

Una interfaz de usuario bien diseñada debe permitir a los no programadores llevar a cabo muchas aplicaciones de principio a fin. Además, el sistema OLP debe tener la capacidad de traducir las tramas y las secuencias de movimiento "enseñadas" por los no programadores en instrucciones textuales del RPL. Estos programas simples pueden mantenerse y embellecerse en formato de RPL por programadores más experimentados. Para los programadores, la disponibilidad del RPL permite el desarrollo de código arbitrario para aplicaciones más complejas.

Modelado en 3-D

Un elemento central en los sistemas OLP es el uso de descripciones gráficas del robot simulado y su célula de trabajo. Para ello se requiere que el robot y todos los dispositivos fijos, piezas y herramientas en la célula de trabajo se modelen como objetos tridimensionales. Para agilizar el desarrollo del programa, es conveniente utilizar modelos CAD de piezas o herramientas que estén directamente disponibles desde el sistema CAD en el que se haya realizado el diseño original. A medida que prevalezcan cada vez más los sistemas CAD en la industria, será más probable que este tipo de datos geométricos esté disponible y listo para usarse. Debido al urgente deseo de tener este tipo de integración

CAD del diseño a la producción, tiene sentido que un sistema OLP contenga un subsistema de modelado CAD o que sea, en sí, una parte de un sistema de diseño de este tipo. Si un sistema OLP va a ser autónomo, debe tener las interfaces apropiadas para transferir modelos desde y hacia sistemas CAD externos; no obstante, hasta un sistema OLP autónomo debería tener cuando menos una simple instalación CAD local para crear rápidamente modelos de elementos no críticos de la célula de trabajo, o para agregar datos específicos del robot a los modelos CAD importados.

Los sistemas OLP por lo común requieren de varias representaciones de formas espaciales. Para muchas operaciones, generalmente existe una descripción analítica exacta de la superficie o volumen; aún así, para poder beneficiarse de la tecnología de visualización a menudo se requiere otra representación. La tecnología actual está bien adaptada a los sistemas en los que la primitiva de visualización subyacente es un polígono plano; sin embargo, aunque la forma de un objeto pueda estar bien representada por una superficie uniforme, la visualización práctica (especialmente para la animación) requiere una representación por facetas. Las acciones gráficas de la interfaz de usuario, como señalar un punto en una superficie, deberían actuar internamente de manera que se especifique un punto en la superficie real, incluso si gráficamente el usuario vea una descripción del modelo por facetas.

Un uso importante de la geometría tridimensional de los modelos de objetos es para la **detección automática de colisiones**; esto es, cuando ocurre una colisión entre los objetos en un entorno simulado, el sistema OLP debería advertir automáticamente al usuario e indicarle exactamente en dónde ocurrió la colisión. Aplicaciones como el ensamblaje pueden implicar muchas “colisiones” deseadas, por lo que es necesario poder informar al sistema que las colisiones entre ciertos objetos son aceptables. También es importante poder generar una advertencia de colisión cuando los objetos pasan dentro de una tolerancia especificada de una verdadera colisión. En la actualidad es difícil detectar con exactitud las colisiones para sólidos generales en 3-D, pero la detección de colisiones para modelos por facetas es bastante práctica.

Emulación cinemática

Un componente imprescindible para mantener la validez del mundo simulado es la emulación fiel de los aspectos geométricos de cada manipulador simulado. Respecto a la cinemática inversa, el sistema OLP puede interactuar con el controlador de robot en dos formas distintas. En primer lugar, el sistema OLP podría sustituir la cinemática inversa del controlador de robot y siempre comunicar las posiciones de éste en el espacio de articulación del mecanismo. La segunda opción es comunicar las ubicaciones cartesianas al controlador de robot y dejar que el controlador utilice la cinemática inversa suministrada por el fabricante para resolver las configuraciones del robot. Casi siempre es preferible usar la segunda opción, sobre todo porque los fabricantes están empezando a construir una calibración estilo *rastro del brazo* en sus robots. Estas técnicas de calibración personalizan la cinemática inversa para cada robot individual. En este caso es conveniente comunicar la información a los controladores de robot a nivel cartesiano.

Estas consideraciones por lo general significan que las funciones de cinemática directa e inversa utilizadas por el simulador deben reflejar las funciones nominales utilizadas en el controlador de robot suministrado por el fabricante del robot. Hay varios detalles de la función cinemática inversa especificada por el fabricante que el simulador de software debe emular. Cualquier algoritmo de cinemática inversa debe hacer elecciones arbitrarias para poder resolver las singularidades. Por ejemplo, cuando la articulación 5 de un robot PUMA 560 se encuentra en su posición cero, los ejes 4 y 6 se alinean y existe una

condición singular. La función de cinemática inversa en el controlador del robot puede despejar la suma de los ángulos de las articulaciones 4 y 6, pero después debe usar una regla arbitraria para elegir valores individuales para las articulaciones 4 y 6. El sistema OLP debe emular cualquier algoritmo que se utilice. Otro ejemplo sería elegir la solución más adecuada cuando existen muchas soluciones alternativas. El simulador debe utilizar el mismo algoritmo que el controlador para poder evitar potenciales errores catastróficos al simular el manipulador real. Una característica útil que se encuentra poco en los controladores de robots es la habilidad de comandar un destino cartesiano y especificar cuál de las posibles soluciones debe utilizar el manipulador. La existencia de esta característica elimina el requerimiento de que el simulador emule el algoritmo de solución elegido; el sistema OLP puede simplemente forzar su elección en el controlador.

Emulación de planeación de rutas

Además de la emulación cinemática para el posicionamiento estático del manipulador, un sistema OLP debería emular con precisión la ruta tomada por el manipulador para moverse a través del espacio. De nuevo, el problema principal es que el sistema OLP necesita simular los algoritmos en el controlador de robot empleado, y dichos algoritmos de planeación y ejecución de rutas varían considerablemente de un fabricante de robots a otro. La simulación de la forma espacial de la ruta tomada es importante para la detección de colisiones entre el robot y su entorno. La simulación de los aspectos temporales de la trayectoria es importante para predecir los tiempos de ciclo de las aplicaciones. Cuando un robot opera en un entorno móvil (por ejemplo, cerca de otro robot) es necesaria la simulación precisa de los atributos temporales de movimiento para predecir las colisiones con exactitud y, en algunos casos, para prever los problemas de comunicación o de sincronización, como el interbloqueo.

Emulación dinámica

El movimiento simulado de los manipuladores puede ignorar los atributos dinámicos si el sistema OLP hace un buen trabajo para emular el algoritmo de planeación de trayectorias del controlador, y si el robot sigue las trayectorias deseadas con errores insignificantes. No obstante, en condiciones de alta velocidad o bajo una carga pesada, los errores de rastreo de trayectorias pueden volverse importantes. Para simular estos errores de rastreo se necesita tanto modelar el comportamiento dinámico del manipulador y de los objetos que mueve, como emular el algoritmo de control utilizado en el controlador del manipulador. En la actualidad existen problemas prácticos para obtener suficiente información de los fabricantes de robots como para que este tipo de simulación dinámica tenga un valor práctico pero, en algunos casos, la simulación dinámica puede lograrse con éxito.

Simulación de múltiples procesos

Algunas aplicaciones industriales involucran a dos o más robots que cooperan en el mismo entorno. Incluso hasta las células de trabajo con un solo robot contienen una banda transportadora, una línea de transferencia, un sistema de visión, o algún otro dispositivo activo con el que el robot debe interactuar. Por esta razón es importante que un sistema OLP pueda simular múltiples dispositivos móviles y otras actividades que involucren el **paralelismo**. Como base para esta capacidad, el lenguaje subyacente en el que se implementa el sistema debería ser un lenguaje de multiprocesamiento. Dicho ambiente hace posible la escritura de programas de control de robots independientes para cada uno de los dos o más robots en una sola célula, y luego se simula la acción de

la célula con los programas ejecutándose concurrentemente. Al agregar primitivas de señalización y espera al lenguaje se permite a los robots interactuar entre sí, de igual forma que lo harían en la aplicación que se está simulando.

Simulación de sensores

Los estudios han demostrado que un gran componente de los programas de robots consiste no solamente de instrucciones de movimiento, sino de inicialización, comprobación de errores, E/S y de otros tipos [7]. Por ende, la habilidad del sistema OLP de ofrecer un entorno que permita la simulación de aplicaciones completas, incluyendo la interacción con sensores, varios tipos de E/S y la comunicación con otros dispositivos se vuelve importante. Un sistema OLP que soporte la simulación de sensores y multiprocesamiento no sólo puede verificar la viabilidad de los movimientos del robot, sino que también puede verificar la porción de comunicación y sincronización del programa del robot.

Traducción del lenguaje al sistema de destino

Una molestia para los usuarios actuales de los robots industriales (y de cualquier otro tipo de automatización programable) es que casi todos los proveedores de dichos sistemas han inventado un lenguaje único para programar su producto. Si un sistema OLP aspira a ser universal en el equipo que puede manejar, debe traducir desde y hacia distintos lenguajes. Una opción para solucionar este problema es elegir un solo lenguaje a utilizar por el sistema OLP y después posprocesar el lenguaje para convertirlo al formato requerido por la máquina de destino. También es conveniente la habilidad de cargar programas que ya existan en las máquinas de destino y llevarlos al sistema OLP.

Dos beneficios potenciales de los sistemas OLP se relacionan directamente con el tema de traducción del lenguaje. La mayoría de los ponentes de los sistemas OLP saben que al tener una interfaz universal que permita a los usuarios programar una variedad de robots, se resuelve el problema de aprender y tratar con varios lenguajes de automatización. Un segundo beneficio se deriva de las consideraciones económicas en escenarios futuros en los que cientos, o tal vez miles de robots, llenan las fábricas. El costo asociado con un poderoso entorno de programación (como un lenguaje y una interfaz gráfica) podría prohibir colocarlo en el sitio de instalación de cada robot. Parece que, en relación con lo económico, tiene sentido colocar un controlador muy simple, “tonto” y económico con cada robot y descargarlo desde un poderoso e “inteligente” sistema OLP que esté ubicado en un entorno de oficina. En consecuencia, el problema general de traducir un programa de aplicación de un poderoso lenguaje universal a un simple lenguaje diseñado para ejecutarse en un procesador económico se vuelve un asunto importante en los sistemas OLP.

Calibración de la célula de trabajo

Una inevitable realidad de un modelo de computadora en cualquier situación del mundo real es la imprecisión en el modelo. Para poder hacer que los programas desarrollados en un sistema OLP puedan utilizarse, los métodos de **calibración de la célula de trabajo** deben ser una parte integral del sistema. La magnitud de este problema varía considerablemente dependiendo de la aplicación; esta variabilidad hace que la programación fuera de línea de algunas tareas sea mucho más práctica que la de otras. Si la mayor parte del trabajo indica que se le debe volver a enseñar al robot real una aplicación para resolver problemas de imprecisión, los sistemas OLP pierden su efectividad.

Muchas aplicaciones implican el frecuente desempeño de acciones relativas a un objeto rígido. Considere por ejemplo la tarea de perforar varios cientos de hoyos en

un mamparo. La ubicación actual del mamparo relativa al robot puede programarse si se utiliza el robot para tomar tres mediciones. Las ubicaciones de todos los hoyos pueden actualizarse automáticamente si están disponibles en coordenadas de piezas de un sistema CAD. En esta situación sólo hay que enseñar estos tres puntos al robot, en lugar de cientos. La mayoría de las tareas implica este tipo de paradigma de "muchas operaciones relativas a un objeto rígido"; por ejemplo, la inserción de componentes en tableros de PC, el enrutamiento, la soldadura por puntos, la soldadura por arco, el acomodo de tarimas, la pintura y el desbarbado.

13.3 EL SIMULADOR 'PILOT'

En esta sección consideraremos uno de esos sistemas de simulación fuera de línea: el sistema 'Pilot' desarrollado por Adept Technology [8]. El sistema Pilot es en realidad una *suite* de tres sistemas de simulación muy relacionados entre sí; aquí veremos la porción de Pilot (conocida como "Pilot/Cell") que se utiliza para simular una célula de trabajo individual en una fábrica. En particular, este sistema es inusual ya que trata de modelar varios aspectos del mundo físico como un medio para liberar de carga al programador del simulador. En esta sección hablaremos de los "algoritmos geométricos" que se utilizan para dar al simulador la capacidad de emular ciertos aspectos de la realidad física.

La necesidad de facilitar el uso impulsa a hacer que el sistema de simulación se comporte como en el mundo físico real. Entre más parecido al mundo real actúe el simulador, más simple se volverá para el usuario el paradigma de la interfaz de usuario, ya que el mundo físico es el que todos conocemos. Al mismo tiempo, debido al intercambio de facilidad de uso por velocidad computacional y otros factores, se ha producido un diseño en el que un "pedazo" específico de realidad se simula, mientras que muchos detalles no.

Pilot se adapta bien como anfitrión (*host*) para una variedad de algoritmos geométricos. La necesidad de modelar varias porciones del mundo real, junto con la de facilitar al usuario su utilización mediante la automatización de cálculos geométricos frecuentes, impulsa a generar dichos algoritmos. Pilot proporciona el entorno en el que pueden aplicarse algunos algoritmos avanzados en los problemas reales que se presentan en la industria.

Una decisión que se hizo desde el comienzo del diseño del sistema de simulación Pilot fue que el *paradigma de programación* debería ser lo más parecido posible a la manera en que se programaría el sistema de robot real. Se proporcionan ciertas herramientas de planeación y optimización de un nivel más alto, pero se consideró que era importante que la interacción de programación básica fuera similar a los sistemas de hardware actuales. Esta decisión ha conducido el desarrollo del producto hacia una ruta sobre la que encontramos la genuina necesidad de varios algoritmos geométricos. Los algoritmos necesarios varían ampliamente, desde los extremadamente simples hasta los considerablemente complejos.

Si se va a programar un simulador para igualar el comportamiento del mundo real, entonces las acciones y reacciones del mundo físico deben modelarse "automáticamente" por el simulador. El objetivo es liberar al usuario de tener que escribir cualquier "código específico de simulación" en el sistema. Como un ejemplo sencillo, si se ordena a la pinza de sujeción del robot que se abra, la pieza que esté sujetando debe caer en respuesta a la gravedad y posiblemente incluso rebotar y quedar en cierto estado estable. Si obligamos al usuario del sistema a especificar estas acciones del mundo real, el simulador no cumpliría su objetivo: ser programado de la misma forma en que funciona el sistema real. Sólo puede lograrse una máxima facilidad de uso cuando el mundo simulado "sabe cómo" comportarse igual que el mundo real sin importunar al usuario.

La mayoría de los sistemas comerciales (si no es que todos) para simular robots u otros mecanismos no tratan directamente con este problema. En su lugar, "permiten" al usuario (en realidad, *obligan* al usuario) a incrustar comandos específicos de la simulación dentro del programa escrito para controlar el dispositivo simulado. Un ejemplo sencillo sería la siguiente secuencia de código:

```
MOVE TO recoger_pieza
CLOSE pinza
affix(pinza, pieza[i]);
MOVE TO colocar_pieza
OPEN pinza
unaffix(pinza, pieza[i]);
```

Aquí se ha obligado al usuario a insertar comandos de "fijación" (affix) y "liberación" (unaffix), los que (respectivamente) hacen que la pieza se mueva con la pinza de sujeción cuando se recoge y que deje de moverse cuando se libera. Si el simulador permite que el robot sea programado en su lenguaje nativo, por lo general ese lenguaje no es lo suficientemente completo como para soportar estos comandos, por lo tanto requiere un segundo grupo de comandos, incluso hasta podría tener una sintaxis distinta para lidiar con las interacciones en el mundo real. Dicho esquema *no* está programado inherentemente "igual que el sistema físico", por lo que debe producir un aumento en la carga de programación para el usuario.

Del ejemplo anterior podemos ver el primer algoritmo geométrico para el que encontramos una necesidad: de la geometría de la pinza de sujeción y las ubicaciones relativas de las piezas, averigüe qué pieza (si es que la hay) se sujetará cuando la pinza de sujeción se cierre y posiblemente cómo se autoalineará la pieza dentro de la pinza. En el caso de Pilot, resolvemos la primera parte de este problema con un algoritmo simple. En casos limitados, la "acción de alineación" de la pieza en la pinza se calcula, pero en general, dichas alineaciones tienen que ser programadas previamente por el usuario del sistema. Por ende podemos ver que Pilot no ha llegado al objetivo definitivo aún, pero ha dado algunos pasos en esa dirección.

Modelado físico y sistemas interactivos

En un sistema de simulación siempre se intercambia la complejidad del modelo en términos de tiempo de cómputo por la precisión de la simulación. En el caso de Pilot y sus objetivos determinados, es muy importante mantener el sistema completamente interactivo. Esto ha conducido a diseñar a Pilot de manera que pueda utilizar varios modelos aproximados; por ejemplo, el uso de aproximaciones cuasi-estáticas en donde un modelo por completo dinámico podría ser más preciso. Aunque parece haber la posibilidad de que pronto puedan aplicarse los modelos "completamente dinámicos" [9], dado el estado actual del hardware de computadora, de los algoritmos dinámicos y de la complejidad de los modelos CAD que los usuarios industriales desean emplear, sentimos que aún hay que hacer estos intercambios.

Algoritmos geométricos para la caída de piezas

En algunos sistemas de alimentación empleados en la práctica industrial, las piezas caen de cierta forma del transportador de alimentación interna hacia una superficie de presentación; luego se utiliza la visión computacional para localizar las piezas que el robot va a tomar. El proceso de diseñar dichos sistemas de automatización con la ayuda de un simulador significa que éste debe ser capaz de predecir la manera en que las piezas van a caer, rebotar y tomar una orientación estable, o *estado estable*.

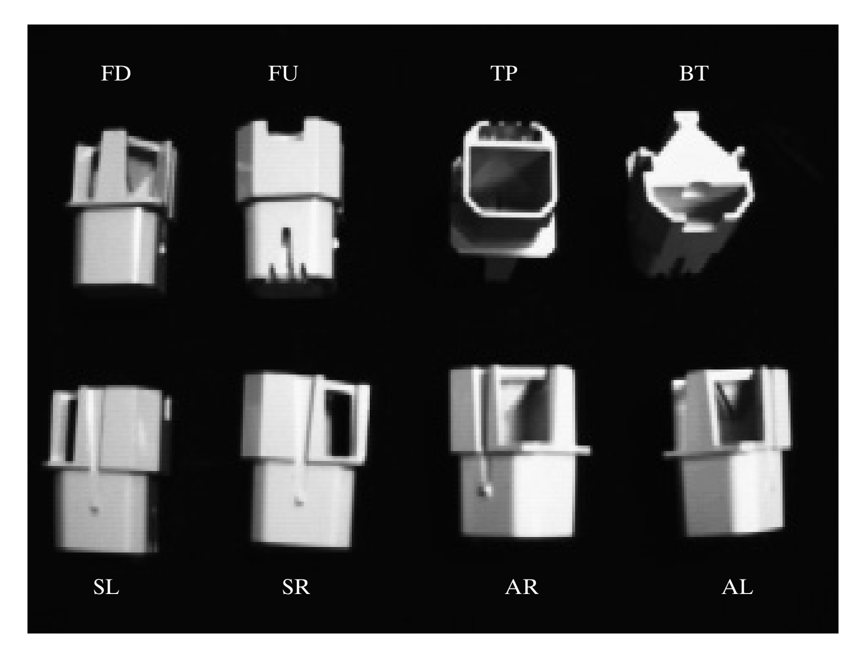


FIGURA 13.1: Los ocho estados estables de la pieza.

Probabilidades de estado estable

Como se informó en la referencia [10], se ha implementado un algoritmo que toma como entrada cualquier forma geométrica (representada por un modelo CAD) y, para esa forma, puede calcular las *N* posibles formas en que puede permanecer estable en una superficie horizontal. A estas formas se les conoce como *estados estables* de la pieza. Además, el algoritmo utiliza un método perturbado cuasi-estático para estimar la probabilidad asociada a cada uno de los *N* estados estables. Hemos realizado experimentos físicos con piezas de muestra para valorar la precisión resultante de la predicción de estado estable.

La figura 13.1 muestra los ocho estados estables de una pieza de prueba específica. Utilizando un robot Adept y un sistema de visión, dejamos caer esta pieza más de 26 mil veces y registramos el estado estable resultante para comparar nuestro algoritmo de predicción de estado estable con la realidad. La tabla 13.1 muestra los resultados para la pieza de prueba, los cuales son característicos de nuestro algoritmo actual; el error de predicción de probabilidad de estado estable varía por lo regular del 5 al 10 por ciento.

Ajuste de las probabilidades como función de la altura de la caída

Evidentemente, si se deja caer una pieza de una pinza de sujeción desde una altura muy pequeña (por ejemplo, 1 mm) sobre una superficie, las probabilidades de los diversos estados estables difieren de las que ocurren cuando la pieza se deja caer desde una altura mayor a cierta *altura crítica*. En Pilot utilizamos probabilidades del algoritmo *estimador de estado estable* cuando se dejan caer piezas desde alturas iguales o mayores que la mayor dimensión de la pieza. Para las alturas de caída por debajo de este valor, las probabilidades se ajustan para tomar en cuenta la orientación inicial de la pieza y la altura de la

TABLA 13.1: Comparación entre probabilidades pronosticadas y reales de estado estable para la pieza de prueba

Estado estable	# real	% real	% pronosticado
FU	1871	7.03%	8.91%
FD	10,600	39.80%	44.29%
TP	648	2.43%	7.42%
BT	33	0.12%	8.19%
SR	6467	24.28%	15.90%
SL	6583	24.72%	15.29%
AR/AL	428	1.61%	0.00%
Total	26,630	100%	100%

caída. El ajuste es tal que, a medida que se llega a una altura de caída infinitesimal, la pieza permanece en su orientación inicial (suponiendo que sea una orientación estable). Ésta es una adición importante al algoritmo de probabilidad general, ya que es común que se liberen piezas a una distancia pequeña sobre una superficie de apoyo.

Simulación del rebote

Las piezas en Pilot están etiquetadas con su coeficiente de restitución; también lo están todas las superficies en las que pueden colocarse las piezas. El producto de estos dos factores se utiliza en una fórmula para predecir qué tan lejos rebotará la pieza cuando se deje caer. Estos detalles son importantes, ya que afectan la forma en que las piezas se esparcen o se agrupan en la simulación de ciertos sistemas de alimentación. Al rebotar, las piezas se esparcen en forma radial, de acuerdo a una distribución uniforme. La distancia de rebote (lejos del punto inicial de contacto) es una cierta función de distribución hasta una distancia máxima, la cual se calcula como una función de la altura de caída (entrada de energía) y los coeficientes de restitución que se apliquen.

Las piezas en Pilot pueden rebotar en forma recursiva de superficie en superficie en ciertos arreglos. También es posible marcar ciertas superficies de tal forma que las piezas no puedan rebotar *fuera* de ellas, sino sólo *dentro* de ellas. Las entidades conocidas como *recipientes* en Pilot tienen esta propiedad; las piezas pueden caer en ellos, pero nunca rebotar hacia afuera.

Simulación de apilamiento y enredamiento

Como una simplificación, las piezas en Pilot siempre descansan en superficies de apoyo planas. Si las piezas se enredan o se apilan una sobre otra, esto se muestra como piezas que se intersecan entre sí (esto es, la intersección booleana de sus volúmenes no estaría vacía). Esto ahorra la enorme cantidad de cálculos que tendrían que realizarse para calcular las diversas formas en que una pieza podría apilarse o enredarse con la geometría de otra pieza.

Las piezas en Pilot están etiquetadas con un *factor de enredamiento*. Por ejemplo, una canica tendría un factor de enredamiento de 0.0 ya que, cuando se dejan caer en una superficie de apoyo, las canicas nunca se apilan o enredan, sino que tienden a esparcirse en la superficie. Por otra parte, piezas como los resortes en espiral podrían tener un factor de enredamiento cercano a 1.0, y es muy probable que se enreden uno con otro.

Cuando una pieza cae y rebota se ejecuta un algoritmo para *encontrar el espacio*, en donde la pieza trata de rebotar hacia un espacio abierto. Sin embargo, la exactitud con la que trate de encontrar un espacio abierto es una función de su factor de enredamiento. Mediante el ajuste de este coeficiente, Pilot puede simular piezas que se dejan caer y se enredan en mayor o menor grado. En la actualidad no existe un algoritmo para calcular automáticamente el factor de enredamiento a partir de la geometría de la pieza; éste es un interesante problema abierto. A través de la interfaz, el usuario de Pilot puede establecer el factor de enredamiento al valor que considere apropiado.

Algoritmos geométricos para sujetar piezas

La mayor parte de la dificultad en programar y utilizar robots tiene que ver con los detalles de programar las ubicaciones de sujeción en las piezas y con el diseño detallado de las pinzas de sujeción. Ésta es un área en la que los algoritmos de planeación adicionales en un sistema simulador podrían tener un gran impacto. En esta sección veremos los algoritmos que se utilizan actualmente en Pilot. Los métodos actuales son bastante simples, por lo que ésta es un área de trabajo continuo.

Cómo calcular qué pieza sujetar

Cuando una herramienta se cierra, o cuando se acciona un efector final de succión, Pilot aplica un algoritmo simple para calcular qué parte (si la hay) debe ser sujetada por el robot. Primero, el sistema averigua qué superficie de apoyo se encuentra inmediatamente debajo de la pinza de sujeción. Después, para todas las piezas en la superficie, busca aquélla cuya caja delimitadora (para el estado estable actual) contenga el TCP (punto central de la herramienta) de la pinza de sujeción. Si más de una pieza cumple con este criterio, entonces selecciona la que se encuentre más cerca.

Cálculo de la ubicación de sujeción predeterminada

Pilot asigna automáticamente una ubicación de sujeción para cada orientación estable pronosticada por el estimador de estado estable descrito antes. El algoritmo actual es simplista, por lo que también se proporciona una interfaz gráfica de usuario para que éste pueda editar y redefinir estos puntos de sujeción. El algoritmo de sujeción actual es una función de la caja delimitadora de la pieza y de la geometría de la pinza de sujeción, que se supone debe ser una pinza de sujeción tipo quijada en paralelo o una ventosa. Además de calcular una ubicación de sujeción predeterminada para cada estado estable, también se calculan un acercamiento predeterminado y una altura de salida.

Cálculo de la alineación de la pieza durante la sujeción

En ciertos casos importantes de la práctica industrial, el diseñador del sistema cuenta con el hecho de que, cuando se acciona el efector final del robot, la pieza capturada se alinea por sí sola en cierta forma con las superficies del efector final. Este efecto puede ser importante para eliminar los pequeños desalineamientos en la presentación de las piezas al robot.

Un efecto bastante real que necesita simularse es que con las pinzas de sujeción tipo ventosa, puede darse el caso de que al aplicar succión, la pieza se "levante" hacia la ventosa en forma tal que altere considerablemente su orientación relativa al efector final. Pilot simula este efecto marcando la geometría de la pieza con una línea vertical alineada con la línea central de la ventosa. La faceta del modelo poligonal de la pieza que se marque es la que se utiliza para calcular la orientación al momento de la sujeción;

la normal de esta faceta se antialínea con la normal de la parte inferior de la ventosa. Al alterar la orientación de la pieza se minimiza la rotación sobre esta línea de marcado (la pieza no gira sobre el eje de la ventosa cuando se sujeta). Sin la simulación de este efecto, el simulador no podría representar en forma realista ciertas estrategias de "recoger y colocar" en las que se utilicen pinzas de sujeción tipo succión.

También hemos implementado un planificador que permite que las piezas giren sobre el eje Z cuando una pinza de sujeción tipo quijada en paralelo se cierra sobre ellas. Este caso es automático sólo para un caso simple; en otras situaciones, el usuario debe enseñar la alineación resultante (es decir, aún estamos esperando un algoritmo que esté más cerca de estar terminado).

Algoritmos geométricos para empujar piezas

Un estilo de empuje de piezas ocurre entre las quijadas de una pinza de sujeción, como se mencionó en la sección anterior. En la práctica industrial actual, las piezas algunas veces son empujadas por mecanismos simples. Por ejemplo, después de que un alimentador de tazón presenta una pieza, ésta podría ser empujada por un actuador lineal justo hacia un ensamblaje que se haya llevado a la celda a través de un sistema transportador de bandeja.

Pilot tiene capacidad para simular el empuje de las piezas: una entidad llamada *barra de empuje*, la cual puede unirse a un cilindro neumático o a un actuador de tornillo de avance en el simulador. Cuando el actuador mueve la barra de empuje a lo largo de una ruta lineal, la superficie delantera de la barra de empuje mueve las piezas. En el futuro se planea que las barras de empuje también puedan agregarse como guías a lo largo de los transportadores o que puedan colocarse en cualquier parte que requiera que el movimiento de las piezas se vea afectado por su presencia. El empuje actual es aún muy simple, pero es suficiente para muchas tareas del mundo real.

Algoritmos geométricos para los transportadores de bandeja

Pilot soporta la simulación de sistemas de transportadores de bandeja en los que las bandejas se mueven a lo largo de rastros compuestos por componentes en línea recta y de corte circular. Puede haber *compuertas* colocadas a lo largo de los rastros en ubicaciones clave, las cuales se abren temporalmente para bloquear una bandeja cuando se le ordena. Además, pueden colocarse *sensores* que detecten el paso de una bandeja en ubicaciones especificadas por el usuario. Dichos sistemas de transportación son comunes en muchos esquemas de automatización.

Conexión de transportadores de bandeja, fuentes y vertederos

Los transportadores de bandeja pueden conectarse entre sí para permitir varios estilos de ramificaciones. Donde dos transportadores "fluyen juntos", se proporciona un esquema simple para evitar colisiones, para hacer que las bandejas del transportador *secundario* sean subordinadas a las bandejas del transportador *principal*. Las bandejas en el transportador secundario esperarán siempre que pueda ocurrir una colisión. En las conexiones de "flujo aparte" se agrega un dispositivo llamado *director* al transportador principal, el cual puede utilizarse para controlar la dirección que tomará una bandeja en la intersección. Las líneas de E/S digitales conectadas al controlador de robot simulado se utilizan para leer sensores, activar compuertas y activar directores.

En los extremos de un transportador de bandejas hay una *fuente* y un *vertedero*. El usuario establece las fuentes para generar bandejas en ciertos intervalos estadísticos. Las bandejas generadas podrían estar vacías o precargadas con piezas o dispositivos

fijos. En el extremo de un transportador de bandeja, éstas (y su contenido) desaparecen en los vertederos. Cada vez que una bandeja entra en un vertedero se registra su tiempo de llegada y su contenido. Estos *registros de vertedero* pueden entonces reproducirse a través de una fuente en cualquier otra parte del sistema. Por ende, una línea de células puede estudiarse en el simulador célula por célula, estableciendo la fuente de la célula $N + 1$ con el registro de vertedero de la célula N.

Empuje de las bandejas

También se implementa el empuje para las bandejas: puede usarse una barra de empuje para propulsar una bandeja fuera de un sistema transportador de bandejas y hacia una célula de trabajo específica. De igual forma, las bandejas pueden empujarse hacia un transportador de bandeja. La actualización de diversas estructuras de datos cuando las bandejas salen de un transportador o entran en uno es una parte automática del código de empuje.

Algoritmos geométricos para los sensores

Se requiere la simulación de varios sistemas de sensores para que el usuario no tenga que escribir código para emular su comportamiento en la célula.

Sensores de proximidad

Pilot soporta la simulación de sensores de proximidad y de otros tipos. En el caso de los sensores de proximidad, el usuario etiqueta el dispositivo con sus rangos mínimo y máximo, y con un umbral. Si un objeto está dentro del rango y más cerca que el umbral, entonces el sensor lo detecta. Para realizar este cálculo en el mundo simulado, se agrega temporalmente un segmento de línea al entorno simulado, uno que se estire del rango mínimo al rango máximo del sensor. Usando un algoritmo de colisiones, el sistema calcula las ubicaciones en las que este segmento de línea se interseca con otra geometría CAD. El punto de intersección más cercano al sensor corresponde al elemento del mundo real que habría detenido el rayo. Una comparación de la distancia a este punto y el umbral proporciona la salida del sensor. Hasta hoy no hacemos uso del ángulo de la superficie encontrada o de sus propiedades de reflectancia, aunque tal vez podrían agregarse esas características en el futuro.

Sistemas de visión 2-D

Pilot simula el desempeño del sistema de visión 2-D de Adept. La manera en que trabaja el sistema de visión simulado está estrechamente relacionada con la manera en que trabaja el sistema de visión real, incluso la manera en que está programado en el lenguaje AIM [11] utilizado por los robots de Adept. Se simulan los siguientes elementos de este sistema de visión:

- La forma y la extensión del campo de vista.
- La distancia de referencia y un simple modelo de enfoque.
- El tiempo requerido para realizar el procesamiento de la visión (aproximado).
- El orden espacial de los resultados en la cola, en caso de que se encuentren muchas piezas en una imagen.
- La habilidad de distinguir las piezas de acuerdo con el estado estable en el que se encuentren.
- La incapacidad de reconocer piezas que se tocan entre sí o se superponen.
- Dentro del contexto de AIM, la habilidad de actualizar las metas del robot con base en los resultados de la visión.

El uso de un sistema de visión está bien integrado con el sistema de programación de robots AIM, por lo que la implementación de este lenguaje en el simulador implica también el de la emulación del sistema de visión. AIM soporta varias instrucciones que facilitan el uso de la visión para guía del robot. El proceso de recoger piezas que se identifiquen visualmente gracias a la indización y el rastreo de los transportadores se logra con cierta facilidad.

Una estructura de datos lleva el registro de cuál superficie de apoyo es la que está observando el sistema de visión. Para todas las piezas apoyadas en esa superficie, calculamos cuáles están dentro del campo del sistema de visión. Descartamos las piezas que estén demasiado cerca o demasiado lejos de la cámara (es decir, desenfocados), y las que toquen piezas adyacentes. De las piezas restantes elegimos las que se encuentren en el estado estable demandado y las colocamos en una lista. Finalmente, esta lista se ordena para emular el que utiliza el sistema de visión de Adept cuando se encuentran varias piezas en una escena.

Sensores inspectores

Se proporciona una clase especial de sensores, llamados *inspectores*. Un inspector se utiliza para proporcionar una salida binaria para cada pieza que esté colocada en frente de él. Las piezas en Pilot pueden etiquetarse con una *proporción de defectos* y los inspectores pueden descartar las piezas defectuosas. Los inspectores juegan el papel de varios sistemas de sensores del mundo real.

Conclusión

Como se mencionó, aunque hay algunos algoritmos geométricos simples actualmente en funcionamiento en el simulador, existe la necesidad de más y mejores algoritmos. En especial nos gustaría investigar la posibilidad de agregar una capacidad de simulación cuasi-estática para predecir el movimiento de los objetos en situaciones en las que los efectos de fricción dominan cualquier efecto inercial. Esto podría usarse para simular las piezas que se empujan o se inclinan debido a diversas acciones de los efectores finales u otros mecanismos de empuje.

13.4 AUTOMATIZACIÓN DE SUBTAREAS EN SISTEMAS OLP

En esta sección mencionaremos brevemente algunas características avanzadas que podrían integrarse en la “línea base” del concepto de sistema OLP que ya presentamos. La mayoría de estas características realizan la planeación automatizada de cierta porción, pequeña, de una aplicación industrial.

Colocación automática del robot

Una de las tareas más básicas que pueden lograrse por medio de un sistema OLP es la determinación de la distribución de la célula de trabajo, de manera que uno o más manipuladores puedan llegar a todos los puntos de trabajo requeridos. La operación de determinar la colocación correcta del robot o de la pieza de trabajo mediante prueba y error se completa mucho más rápido en un mundo simulado que en la célula física. Una característica avanzada que automatiza la búsqueda de ubicaciones factibles para un robot o una pieza de trabajo logra avanzar un paso más para reducir la carga sobre el usuario.

La colocación automática puede calcularse mediante una búsqueda directa o (algunas veces) mediante técnicas de búsqueda guiadas por heurística. La mayoría de los robots se montan nivelados en el piso (o techo) y tienen la primera articulación giratoria

perpendicular al piso, por lo que generalmente no se necesita más que hacer una búsqueda por escalado del espacio tridimensional de la colocación del robot. La búsqueda podría optimizar ciertos criterios o podría detenerse al localizar la primera colocación viable del robot o de la pieza. La viabilidad se define como la habilidad de llegar libre de colisiones a todos los puntos de trabajo (o tal vez se le pueda dar una definición aún más sólida). Un criterio razonable para maximizar podría ser cierta forma de *medida de manipulabilidad*, como se discutió en el capítulo 8. En la referencia [12] se ha descrito una implementación que utiliza una medida similar de manipulabilidad. El resultado de dicha colocación automática es una célula en la que el robot puede llegar a todos sus puntos de trabajo en configuraciones *bien acondicionadas*.

Evasión de colisiones y optimización de rutas

La investigación sobre la planeación de rutas libres de colisiones [13, 14] y la planeación de rutas óptimas [15, 16] genera candidatos naturales para incluirse en un sistema OLP. Algunos problemas relacionados que tienen un alcance menor y un espacio de búsqueda más reducido son también de interés. Por ejemplo, considere el problema de utilizar un robot con seis grados de libertad para una tarea de soldadura por arco cuya geometría especifica sólo cinco grados de libertad. Puede utilizarse la planeación automática del grado redundante de libertad para evitar colisiones y singularidades del robot [17].

Planeación automática del movimiento coordinado

En muchas situaciones de soldadura por arco, los detalles del proceso requieren que se mantenga cierta relación entre la pieza de trabajo y el vector de gravedad durante la tarea de soldado. Esto produce un sistema de orientación con dos o tres grados de libertad en el que se monta la pieza, operando simultáneamente con el robot y en forma coordinada. En un sistema de este tipo, podría haber nueve o más grados de libertad para coordinar. Dichos sistemas por lo común se programan actualmente mediante el uso de técnicas de dispositivo de enseñanza. Sería muy valioso un sistema de planeación que pudiera sintetizar automáticamente los movimientos coordinados para dicho sistema [17, 18].

Simulación del control de fuerza

En un mundo simulado en el que los objetos se representan por sus superficies, es posible investigar la simulación de estrategias de control de fuerza del manipulador. Esta tarea involucra el difícil problema de modelar ciertas propiedades de la superficie y expandir el simulador dinámico para lidiar con las restricciones impuestas por varias situaciones de contacto. En dicho entorno, sería posible evaluar la viabilidad de varias operaciones de ensamblaje controlado por fuerza [19].

Programación automática

Además de los problemas geométricos existentes en la programación de robots, a menudo existen dificultades de programación y comunicación. Esto ocurre especialmente si expandimos la simulación más allá de una sola célula de trabajo a un grupo de células. Algunos sistemas de simulación de tiempo discreto ofrecen la simulación abstracta de dichos sistemas [20], pero pocos ofrecen algoritmos de planeación. La planeación de itinerarios para los procesos que interactúan es un problema difícil y un área de investigación [21, 22]. Un sistema OLP serviría como un medio de prueba ideal para dicha investigación y sería mejorado inmediatamente por cualquier algoritmo útil en esta área.

Valoración automática de errores y tolerancias

Un sistema OLP podría recibir algunas de las capacidades descritas en trabajos recientes para modelar fuentes de error de posicionamiento y el efecto de los datos provenientes de sensores imperfectos [23, 24]. El modelo real podría diseñarse para incluir varios límites de error e información de tolerancia, y el sistema podría evaluar las probabilidades de éxito de varias tareas de posicionamiento o de ensamblaje. El sistema podría sugerir el uso y la colocación de sensores para corregir problemas potenciales.

Los sistemas de programación fuera de línea son útiles en las aplicaciones industriales actuales, y pueden servir como base para la investigación y el desarrollo continuos de la robótica. Una gran innovación en el desarrollo de sistemas OLP es llenar el hueco existente entre los sistemas programados explícitamente que están disponibles actualmente, y los sistemas a nivel de tarea del mañana.

BIBLIOGRAFÍA

[1] J. Craig, "Issues in the Design of Off-Line Programming Systems", *International Symposium of Robotics Research*, R. Bolles y B. Roth, editores, MIT Press, Cambridge, MA, 1988.

[2] J. Craig, "Geometric Algorithms in AdeptRAPID", *Robotics: The Algorithmic Perspective: 1998 WAFR*, P. Agarwal, L. Kavraki y M. Mason, editores, AK Peters, Natick, MA, 1988.

[3] R. Goldman, *Design of an Interactive Manipulator Programming Environment*, UMI Research Press, Ann Arbor, MI, 1985.

[4] S. Mujtaba y R. Goldman, "AL User´s Manual", 3a edición, Stanford Department of Computer Science, Reporte Núm. STAN-CS-81-889, diciembre de 1981.

[5] T. Lozano-Perez, "Spatial Planning: A Configuration Space Approach", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-11, 1983.

[6] B. Shimano, C. Geschke y C. Spalding, "VAL-II: A Robot Programming Language and Control System", *SME Robots VIII Conference*, Detroit, junio de 1984.

[7] R. Taylor, P. Summers y J. Meyer, "AML: A Manufacturing Language", *International Journal of Robotics Research*, vol 1, núm. 3, otoño de 1982.

[8] Adept Technology, Inc., "The Pilot User's Manual", disponible de Adept Technology Inc., Livermore, CA, 2001.

[9] B. Mirtich y J. Canny, "Impulse Based Dynamic Simulation of Rigid Bodies", *Symposium on Interactive 3D Graphics*, ACM Press, Nueva York, 1995.

[10] B. Mirtich, Y. Zhuang, K. Goldberg y otros, "Estimating Pose Statistics for Robotic Part Feeders", *Proceedings of the IEEE Robotics and Automation Conference*, Minneapolis, abril de 1996.

[11] Adept Technology, Inc., "AIM Manual", disponible de Adept Technology Inc., San Jose, CA, 2002.

[12] B. Nelson, K. Pedersen y M. Donath, "Locating Assembly Tasks in a Manipulator's Workspace", *IEEE Conference on Robotics and Automation*, Raleigh, NC, abril de 1987.

[13] T. Lozano-Perez, "A Simple Motion Planning Algorithm for General Robot Manipulators", *IEEE Journal of Robotics and Automation*, vol. RA-3, núm. 3, junio de 1987.

[14] R. Brooks, "Solving the Find-Path Problem by Good Representation of Free Space", *IEEE Transaction on Systems, Man, and Cybernetics*, SMC-13:190-197, 1983.

[15] J. Bobrow, S. Dubowsky y J. Gibson, "On the Optimal Control of Robotic Manipulators with Actuator Constraints", *Proceedings of the American Control Conference,* junio de 1983.

[16] K. Shin y N. McKay, "Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints", *IEEE Transactions on Automatic Control*, junio de 1985.

[17] J. J. Craig, "Coordinated Motion of Industrial Robots and 2-DOF Orienting Tables", *Proceedings of the 17th International Symposium on Industrial Robots*, Chicago, abril de 1987.

[18] S. Ahmad y S. Luo, "Coordinated Motion Control of Multiple Robotic Devices for Welding and Redundancy Coordination through Constrained Optimization in Cartesian Space", *Proceedings of the IEEE Conference on Robotics and Automation*, Filadelfia, 1988.

[19] M. Peshkin y A. Sanderson, "Planning Robotic Manipulation Strategies for Sliding Objects", *IEEE Conference on Robotics and Automation*, Raleigh, NC, abril de 1987.

[20] E. Russel, "Building Simulation Models with Simcript II.5", C.A.C.I., Los Angeles, 1983.

[21] A. Kusiak y A. Villa, "Architectures of Expert Systems for Scheduling Flexible Manufacturing Systems", *IEEE Conference on Robotics and Automation*, Raleigh, NC, abril de 1987.

[22] R. Akella y B. Krogh, "Hierarchical Control Structures for Multicell Flexible Assembly System Coordination", *IEEE Conference on Robotics and Automation*, Raleigh, NC, abril de 1987.

[23] R. Smith, M. Self y P. Cheeseman, "Estimating Uncertain Spatial Relationships in Robotics", *IEEE Conference on Robotics and Automation*, Raleigh, NC, abril de 1987.

[24] H. Durrant-Whyte, "Uncertain Geometry in Robotics", *IEEE Conference on Robotics and Automation, Raleigh*, NC, abril de 1987.

EJERCICIOS

13.1 [10] Defina en una o dos oraciones lo que es detección de colisiones, evasión de colisiones y planeación de rutas libre de colisiones.

13.2 [10] Defina en una o dos oraciones lo que es modelo del entorno, emulación de planeación de rutas y emulación dinámica.

13.3 [10] Defina en una o dos oraciones lo que es colocación automática de robots, rutas óptimas en tiempo y análisis de propagación de errores.

13.4 [10] Defina en una o dos oraciones RPL, TLP y OLP.

13.5 [10] Defina en una o dos oraciones lo que es calibración, movimiento coordinado y programación automática.

13.6 [20] Genere un gráfico que indique cómo ha aumentado la habilidad gráfica de las computadoras durante los últimos diez años (tal vez en términos del número de vectores dibujados por segundo, por cada 10,000 dólares de hardware).

13.7 [20] Haga una lista de tareas que estén caracterizadas por "muchas operaciones relativas a un objeto rígido" y que por lo tanto sean candidatas para la programación fuera de línea.

13.8 [20] Escriba las ventajas y desventajas de utilizar un sistema de programación que mantenga un modelo detallado del entorno en forma interna.

EJERCICIO DE PROGRAMACIÓN (PARTE 13)

1. Considere la forma plana de una barra con tapones terminales. A esta forma la llamaremos "cápsula". Escriba una rutina que, dada la ubicación de dos de esas cápsulas, calcule si se están tocando. Tenga en cuenta que todos los puntos superficiales de una cápsula son equidistantes desde un segmento de una sola línea que podría llamarse su "espina dorsal".

2. Introduzca un objeto en forma de cápsula cerca de su manipulador simulado y pruebe si hay colisiones a medida que mueve el manipulador sobre una ruta. Use vínculos en forma de cápsulas para el manipulador. Reporte cualquier colisión.

3. Si el tiempo y la computadora lo permiten, escriba rutinas para describir gráficamente las cápsulas que forman su manipulador y los obstáculos en el espacio de trabajo.

APÉNDICE A

Entidades trigonométricas

Fórmulas para la rotación sobre los ejes principales con base en θ:

$$R_X(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\operatorname{sen}\theta \\ 0 & \operatorname{sen}\theta & \cos\theta \end{bmatrix}, \quad (A.1)$$

$$R_Y(\theta) = \begin{bmatrix} \cos\theta & 0 & \operatorname{sen}\theta \\ 0 & 1 & 0 \\ -\operatorname{sen}\theta & 0 & \cos\theta \end{bmatrix}, \quad (A.2)$$

$$R_Z(\theta) = \begin{bmatrix} \cos\theta & -\operatorname{sen}\theta & 0 \\ \operatorname{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (A.3)$$

Identidades relacionadas con la naturaleza periódica del seno y el coseno:

$$\operatorname{sen}\theta = -\operatorname{sen}(-\theta) = -\cos(\theta + 90^\circ) = \cos(\theta - 90^\circ),$$
$$\cos\theta = \cos(-\theta) = \operatorname{sen}(\theta + 90^\circ) = -\operatorname{sen}(\theta - 90^\circ). \quad (A.4)$$

El seno y el coseno para la suma o diferencia de los ángulos θ_1 y θ_2:

$$\cos(\theta_1 + \theta_2) = c_{12} = c_1c_2 - s_1s_2,$$
$$\operatorname{sen}(\theta_1 + \theta_2) = s_{12} = c_1s_2 + s_1c_2, \quad (A.5)$$
$$\cos(\theta_1 - \theta_2) = c_1c_2 + s_1s_2,$$
$$\operatorname{sen}(\theta_1 - \theta_2) = s_1c_2 - c_1s_2.$$

La suma de los cuadrados del seno y coseno del mismo ángulo es la unidad:

$$c^2\theta + s^2\theta = 1. \quad (A.6)$$

Si los ángulos de un triángulo se llaman a, b y c, en donde el ángulo a es el lado opuesto de A, y así sucesivamente, entonces la "ley de los cosenos" es

$$A^2 = B^2 + C^2 - 2BC\cos a. \quad (A.7)$$

La sustitución de la "tangente del medio ángulo":

$$u = \tan\frac{\theta}{2},$$
$$\cos\theta = \frac{1 - u^2}{1 + u^2}, \quad (A.8)$$
$$\operatorname{sen}\theta = \frac{2u}{1 + u^2}.$$

Para girar un vector Q sobre un vector unitario $\hat{K}$ con base en θ, se utiliza la **fórmula de Rodrigues**:

$$Q' = Q\cos\theta + \operatorname{sen}\theta(\hat{K} \times Q) + (1 - \cos\theta)(\hat{K} \cdot \hat{Q})\hat{K}. \quad \text{(A.9)}$$

En el apéndice B puede consultar las matrices de rotación equivalentes para las convenciones de los 24 conjuntos de ángulos, y en el apéndice C puede consultar algunas identidades de cinemática inversa.

A P É N D I C E B

Convenciones de los 24 conjuntos de ángulos

Los 12 conjuntos de ángulos de Euler se dan con base en

$$R_{X'Y'Z'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\beta c\gamma & -c\beta s\gamma & s\beta \\ s\alpha s\beta c\gamma + c\alpha s\gamma & -s\alpha s\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta \\ -c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha s\beta s\gamma + s\alpha c\gamma & c\alpha c\beta \end{bmatrix},$$

$$R_{X'Z'Y'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\beta c\gamma & -s\beta & c\beta s\gamma \\ c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma \\ s\alpha s\beta c\gamma - c\alpha s\gamma & s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma \end{bmatrix},$$

$$R_{Y'X'Z'}(\alpha,\beta,\gamma)=\begin{bmatrix} s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma & s\alpha c\beta \\ c\beta s\gamma & c\beta c\gamma & -s\beta \\ c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha c\beta \end{bmatrix},$$

$$R_{Y'Z'X'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\alpha c\beta & -c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha s\beta s\gamma + s\alpha c\gamma \\ s\beta & c\beta c\gamma & -c\beta s\gamma \\ -s\alpha c\beta & s\alpha s\beta c\gamma + c\alpha s\gamma & -s\alpha s\beta s\gamma + c\alpha c\gamma \end{bmatrix},$$

$$R_{Z'X'Y'}(\alpha,\beta,\gamma)=\begin{bmatrix} -s\alpha s\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta & s\alpha s\beta c\gamma + c\alpha s\gamma \\ c\alpha s\beta s\gamma + s\alpha c\gamma & c\alpha c\beta & -c\alpha s\beta c\gamma + s\alpha s\gamma \\ -c\beta s\gamma & s\beta & c\beta c\gamma \end{bmatrix},$$

$$R_{Z'Y'X'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & -s\alpha s\beta s\gamma + c\alpha c\gamma & -s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix},$$

$$R_{X'Y'X'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\beta & s\beta s\gamma & s\beta c\gamma \\ s\alpha s\beta & -s\alpha c\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta c\gamma - c\alpha s\gamma \\ -c\alpha s\beta & c\alpha c\beta s\gamma + s\alpha c\gamma & c\alpha c\beta c\gamma - s\alpha s\gamma \end{bmatrix},$$

$$R_{X'Z'X'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\beta & -s\beta c\gamma & s\beta s\gamma \\ c\alpha s\beta & c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma \\ s\alpha s\beta & s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma \end{bmatrix},$$

$$R_{Y'X'Y'}(\alpha,\beta,\gamma)=\begin{bmatrix} -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta & s\alpha c\beta c\gamma + c\alpha s\gamma \\ s\beta s\gamma & c\beta & -s\beta c\gamma \\ -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta & c\alpha c\beta c\gamma - s\alpha s\gamma \end{bmatrix},$$

$$R_{Y'Z'Y'}(\alpha,\beta,\gamma) = \begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha s\beta & c\alpha c\beta s\gamma + s\alpha c\gamma \\ s\beta c\gamma & c\beta & s\beta s\gamma \\ -s\alpha c\beta c\gamma - c\alpha s\gamma & s\alpha s\beta & -s\alpha c\beta s\gamma + c\alpha c\gamma \end{bmatrix},$$

$$R_{Z'X'Z'}(\alpha,\beta,\gamma) = \begin{bmatrix} -s\alpha c\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta c\gamma - c\alpha s\gamma & s\alpha s\beta \\ c\alpha c\beta s\gamma + s\alpha c\gamma & c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha s\beta \\ s\beta s\gamma & s\beta c\gamma & c\beta \end{bmatrix},$$

$$R_{Z'Y'Z'}(\alpha,\beta,\gamma) = \begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix}.$$

Los 12 conjuntos de ángulos fijos se dan con base en

$$R_{XYZ}(\gamma,\beta,\alpha) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix},$$

$$R_{XZY}(\gamma,\beta,\alpha) = \begin{bmatrix} c\alpha c\beta & -c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha s\beta s\gamma + s\alpha c\gamma \\ s\beta & c\beta c\gamma & -c\beta s\gamma \\ -s\alpha c\beta & s\alpha s\beta c\gamma + c\alpha s\gamma & -s\alpha s\beta s\gamma + c\alpha c\gamma \end{bmatrix},$$

$$R_{YXZ}(\gamma,\beta,\alpha) = \begin{bmatrix} -s\alpha s\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta & s\alpha s\beta c\gamma + c\alpha s\gamma \\ c\alpha s\beta s\gamma + s\alpha c\gamma & c\alpha c\beta & -c\alpha s\beta c\gamma + s\alpha s\gamma \\ -c\beta s\gamma & s\beta & c\beta c\gamma \end{bmatrix},$$

$$R_{YZX}(\gamma,\beta,\alpha) = \begin{bmatrix} c\beta c\gamma & -s\beta & c\beta s\gamma \\ c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma \\ s\alpha s\beta c\gamma - c\alpha s\gamma & s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma \end{bmatrix},$$

$$R_{ZXY}(\gamma,\beta,\alpha) = \begin{bmatrix} s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma & s\alpha c\beta \\ c\beta s\gamma & c\beta c\gamma & -s\beta \\ c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha c\beta \end{bmatrix},$$

$$R_{ZYX}(\gamma,\beta,\alpha) = \begin{bmatrix} c\beta c\gamma & -c\beta s\gamma & s\beta \\ s\alpha s\beta c\gamma + c\alpha s\gamma & -s\alpha s\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta \\ -c\alpha s\beta c\gamma + s\alpha s\gamma & c\alpha s\beta s\gamma + s\alpha c\gamma & c\alpha c\beta \end{bmatrix},$$

$$R_{XYX}(\gamma,\beta,\alpha) = \begin{bmatrix} c\beta & s\beta s\gamma & s\beta c\gamma \\ s\alpha s\beta & -s\alpha c\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta c\gamma - c\alpha s\gamma \\ -c\alpha s\beta & c\alpha c\beta s\gamma + s\alpha c\gamma & c\alpha c\beta c\gamma - s\alpha s\gamma \end{bmatrix},$$

$$R_{XZX}(\gamma,\beta,\alpha) = \begin{bmatrix} c\beta & -s\beta c\gamma & s\beta s\gamma \\ c\alpha s\beta & c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma \\ s\alpha s\beta & s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma \end{bmatrix},$$

$$R_{YXY}(\gamma,\beta,\alpha) = \begin{bmatrix} -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta & s\alpha c\beta c\gamma + c\alpha s\gamma \\ s\beta s\gamma & c\beta & -s\beta c\gamma \\ -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta & c\alpha c\beta c\gamma - s\alpha s\gamma \end{bmatrix},$$

$$R_{YZY}(\gamma, \beta, \alpha) = \begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha s\beta & c\alpha c\beta s\gamma + s\alpha c\gamma \\ s\beta c\gamma & c\beta & s\beta s\gamma \\ -s\alpha c\beta c\gamma - c\alpha s\gamma & s\alpha s\beta & -s\alpha c\beta s\gamma + c\alpha c\gamma \end{bmatrix},$$

$$R_{ZXZ}(\gamma, \beta, \alpha) = \begin{bmatrix} -s\alpha c\beta s\gamma + c\alpha c\gamma & -s\alpha c\beta c\gamma - c\alpha s\gamma & s\alpha s\beta \\ c\alpha c\beta s\gamma + s\alpha c\gamma & c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha s\beta \\ s\beta s\gamma & s\beta c\gamma & c\beta \end{bmatrix},$$

$$R_{ZYZ}(\gamma, \beta, \alpha) = \begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix}.$$

APÉNDICE C

Algunas fórmulas de cinemática inversa

La ecuación individual

$$\operatorname{sen}\theta = a \tag{C.1}$$

tiene dos soluciones, dadas por

$$\theta = \pm \mathrm{Atan2}(\sqrt{1-a^2}, a). \tag{C.2}$$

De igual forma, dada la ecuación

$$\cos\theta = b, \tag{C.3}$$

existen dos soluciones:

$$\theta = \mathrm{Atan2}(b, \pm\sqrt{1-b^2}). \tag{C.4}$$

Si se dan (C.1) y (C.3), entonces existe una solución única dada por

$$\theta = \mathrm{Atan2}(a, b). \tag{C.5}$$

La ecuación trascendental

$$a\cos\theta + b\operatorname{sen}\theta = 0 \tag{C.6}$$

tiene las siguientes dos soluciones:

$$\theta = \mathrm{Atan2}(a, -b) \tag{C.7}$$

y

$$\theta = \mathrm{Atan2}(-a, b). \tag{C.8}$$

La ecuación

$$a\cos\theta + b\operatorname{sen}\theta = c, \tag{C.9}$$

que resolvimos en la sección 4.5 con las sustituciones de la tangente del medio ángulo, también se resuelve mediante

$$\theta = \mathrm{Atan2}(b, a) \pm \mathrm{Atan2}(\sqrt{a^2+b^2-c^2}, c). \tag{C.10}$$

El conjunto de ecuaciones

$$\begin{aligned} a\cos\theta - b\operatorname{sen}\theta &= c, \\ a\operatorname{sen}\theta + b\cos\theta &= d, \end{aligned} \tag{C.11}$$

que se resolvió en la sección 4.4, también se resuelve mediante

$$\theta = \mathrm{Atan2}(ad - bc,\ ac + bd). \tag{C.12}$$

Soluciones de ejercicios seleccionados

CAPÍTULO 2 EJERCICIOS SOBRE DESCRIPCIONES ESPACIALES Y TRANSFORMACIONES

2.1)

$$R = ROT(\hat{x}, \phi)ROT(\hat{z}, \theta)$$
$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\phi & -S\phi \\ 0 & S\phi & C\phi \end{bmatrix} \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} C\theta & -S\theta & 0 \\ C\phi S\theta & C\phi C\theta & -S\phi \\ S\phi S\theta & S\phi C\theta & C\phi \end{bmatrix}$$

2.12) La velocidad es un "vector libre" y se ve afectada sólo por la rotación, no por la traslación:

$${}^{A}V = {}^{A}_{B}R\ {}^{B}V = \begin{bmatrix} 0.866 & -0.5 & 0 \\ 0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 10 \\ 20 \\ 30 \end{bmatrix}$$
$${}^{A}V = \begin{bmatrix} -1.34 & 22.32 & 30.0 \end{bmatrix}^T$$

2.27)

$${}^{A}_{B}T = \begin{bmatrix} -1 & 0 & 0 & 3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.33)

$${}^{B}_{C}T = \begin{bmatrix} -0.866 & -0.5 & 0 & 3 \\ 0 & 0 & +1 & 0 \\ -0.5 & 0.866 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

CAPÍTULO 3 EJERCICIOS SOBRE CINEMÁTICA DE MANIPULADORES

3.1)

α_{i-1}	a_{i-1}	d_i
0	0	0
0	L_1	0
0	L_2	0

$$ {}^0_1T = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^1_2T = \begin{bmatrix} C_2 & -S_2 & 0 & L_1 \\ S_2 & C_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^2_3T = \begin{bmatrix} C_3 & -S_3 & 0 & L_2 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^0_3T = {}^0_1T\,{}^1_2T\,{}^2_3T = \begin{bmatrix} C_{123} & -S_{123} & 0 & L_1C_1 + L_2C_{12} \\ S_{123} & C_{123} & 0 & L_1S_1 + L_2S_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

en donde

$$ C_{123} = \cos(\theta_1 + \theta_2 + \theta_3) $$
$$ S_{123} = \operatorname{sen}(\theta_1 + \theta_2 + \theta_3) $$

3.8) Cuando $\{G\} = \{T\}$, tenemos que

$$ {}^B_WT\,{}^W_TT = {}^B_ST\,{}^S_GT $$

Por lo que

$$ {}^W_TT = {}^B_WT^{-1}\,{}^B_ST\,{}^S_GT $$

CAPÍTULO 4 EJERCICIOS SOBRE CINEMÁTICA INVERSA DE MANIPULADORES

4.14) No. El método de Pieper proporciona la solución de forma cerrada para cualquier manipulador 3-DOF. (Vea su tesis para todos los casos).
4.18) 2
4.22) 1

CAPÍTULO 5 EJERCICIOS SOBRE JACOBIANOS: VELOCIDADES Y FUERZAS ESTÁTICAS

5.1) El jacobiano en la trama {0} es

$$ {}^\circ J(\theta) = \begin{bmatrix} -L_1S_1 - L_2S_{12} & -L_2S_{12} \\ L_1C_1 + L_2C_{12} & L_2C_{12} \end{bmatrix} $$

$$ \begin{aligned} \text{DET}({}^\circ J(\theta)) &= -(L_2C_{12})(L_1S_1 + L_2S_{12}) + (L_2S_{12})(L_1C_1 + L_2C_{12}) \\ &= -L_1L_2S_1C_{12} - L_2^2S_{12}C_{12} + L_1L_2C_1S_{12} + L_2^2S_{12}C_{12} \\ &= L_1L_2C_1S_{12} - L_1L_2S_1C_{12} = L_1L_2(C_1S_{12} - S_1C_{12}) \\ &= L_1L_2S_2 \end{aligned} $$

$\therefore$ el mismo resultado que cuando se empieza con ${}^3J(\theta)$, a saber, las configuraciones singulares son $\theta_2 = 0°$ o $180°$.

5.8) El jacobiano de este manipulador de 2 vínculos es

$$^{3}J(\theta) = \begin{bmatrix} L_1 S_2 & 0 \\ L_1 C_2 + L_2 & L_2 \end{bmatrix}$$

Existe un punto isotrópico si

$$^{3}J = \begin{bmatrix} L_2 & 0 \\ 0 & L_2 \end{bmatrix}$$

Por lo tanto,

$$L_1 S_2 = L_2$$
$$L_1 C_2 + L_2 = 0$$

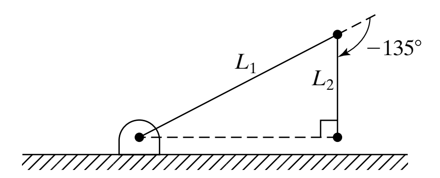
Ahora, $S_2^2 + C_2^2 = 1$, por lo que $\left(\frac{L_2}{L_1}\right)^2 + \left(\frac{-L_2}{L_1}\right)^2 = 1.$

o $L_1^2 = 2L_2^2 \rightarrow L_1 = \sqrt{2}L_2.$

Bajo esta condición, $S_2 = \frac{1}{\sqrt{2}} = \pm.707$

y $C_2 = -.707.$

$\therefore$ Existe un punto isotrópico si $L_1 = \sqrt{2}L_2$, y en ese caso existe cuando $\theta_2 = \pm 135°$.



En esta configuración, el manipulador se ve momentáneamente como un manipulador cartesiano.

5.13)

$$\tau = {}^{\circ}J^T(\theta)^{\circ}F$$

$$\tau = \begin{bmatrix} -L_1 S_1 - L_2 S_{12} & L_1 C_1 + L_2 C_{12} \\ -L_2 S_{12} & L_2 C_{12} \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix}$$

$$\tau_1 = -10 S_1 L_1 - 10 L_2 S_{12}$$

$$\tau_2 = -10 L_2 S_{12}$$

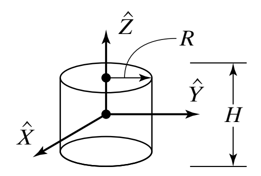
CAPÍTULO 6 EJERCICIOS SOBRE DINÁMICA DE MANIPULADORES

6.1) Use la ecuación (6.17), pero escrita en forma polar, ya que es más sencillo. Por ejemplo, para I_{zz},

$$I_{zz} = \int_{-H/2}^{H/2} \int_0^{2\pi} \int_0^R (x^2 + y^2) p\, r\, dr\, d\theta\, dz$$

$$x = R\cos\theta, \quad y = R\,\text{sen}\,\theta, \quad x^2 + y^2 = R^2(r^2)$$

$$I_{zz} = \int_{-H/2}^{H/2} \int_0^{2\pi} \int_0^R p r^3 dr\, d\theta dz$$



$$I_{zz} = \frac{\pi}{2} R^4 H p, \text{VOLUMEN} = \pi r^2 H$$

$$\therefore \quad \text{Masa} = M = p\pi r^2 H \quad \therefore \quad \boxed{I_{zz} = \tfrac{1}{2} M R^2}$$

De manera similar (sólo que más difícil) se tiene

$$\boxed{I_{xx} = I_{yy} = \tfrac{1}{4} M R^2 + \tfrac{1}{12} M H^2}$$

De la simetría (o integración),

$$\boxed{I_{xy} = I_{xz} = I_{yz} = 0}$$

$\therefore$

$$c_I = \begin{bmatrix} \frac{1}{4}MR^2 + \frac{1}{12}MH^2 & 0 & 0 \\ 0 & \frac{1}{4}MR^2 + \frac{1}{12}MH^2 & 0 \\ 0 & 0 & \frac{1}{2}MR^2 \end{bmatrix}$$

6.12) $\theta_1(t) = Bt + ct^2$, por lo que

$$\dot{\theta}_1 = B + 2ct, \ddot{\theta} = 2c$$

entonces

$$^1\dot{\omega}_1 = \ddot{\theta}_1 \hat{z}_1 = 2c\hat{z}_1 = \begin{bmatrix} 0 \\ 0 \\ 2c \end{bmatrix}$$

$$^1\dot{v}_{c1} = \begin{bmatrix} 0 \\ 0 \\ 2c \end{bmatrix} \otimes \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \otimes \left(\begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \otimes \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \right)$$

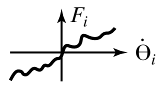
$$= \begin{bmatrix} 0 \\ 4c \\ 0 \end{bmatrix} + \begin{bmatrix} -2\dot{\theta}_1^2 \\ 0 \\ 0 \end{bmatrix}$$

$$^1\dot{v}_{c1} = \begin{bmatrix} -2(B + 2ct)^2 \\ 4c \\ 0 \end{bmatrix}$$

6.18) Cualquier $F(\theta, \dot{\theta})$ razonable probablemente tenga la propiedad de que la fuerza de fricción (o momento de torsión) en la articulación i depende sólo de la velocidad de la articulación i, es decir,

$$F(\theta, \dot{\theta}) = [f_1(\theta, \dot{\theta}_1) \quad F_2(\theta, \dot{\theta}_2) \quad \ldots . F_N(\theta, \dot{\theta}_N)]^T$$

Además, cada $f_i(\theta, \dot{\theta}_i)$ debe ser "pasiva"; es decir, la función debe encontrarse en los cuadrantes primero y tercero.



** ¡Solución escrita a la luz de una vela como secuela de un terremoto; escala 7.0, el 17 de octubre de 1989!

CAPÍTULO 7 EJERCICIOS SOBRE GENERACIÓN DE TRAYECTORIAS

7.1) Se requieren tres polinomios cúbicos para conectar un punto inicial, dos puntos vía y un punto de destino; esto es, tres para cada articulación, para un total de 18 polinomios cúbicos. Cada polinomio cúbico tiene cuatro coeficientes, por lo que se almacenan 72 coeficientes.

7.17) Por diferenciación,

$$\dot{\theta}(t) = 180t - 180t^2$$
$$\ddot{\theta}(t) = 180 - 360t$$

entonces, evaluando en $t = 0$ y $t = 1$, tenemos que

$$\theta(0) = 10 \quad \dot{\theta}(0) = 0 \quad \ddot{\theta}(0) = 180$$
$$\theta(1) = 40 \quad \dot{\theta}(1) = 0 \quad \ddot{\theta}(1) = -180$$

CAPÍTULO 8 EJERCICIOS SOBRE DISEÑO DEL MECANISMO DE UN MANIPULADOR

8.3) Utilizando la ecuación (8.1) tenemos que

$$L = \sum_{i=1}^{3}(a_{i-1} + d_i) = (0 + 0) + (0 + 0) + [0 + (U - L)] = U - L$$

$$W = \frac{4}{3}\pi U^3 - \frac{4}{3}\pi L^3 = \frac{4}{3}\pi(U^3 - L^3) \left\{ \begin{array}{c} \text{una} \\ \text{“esfera”} \\ \text{hueca} \end{array} \right.$$

$$\therefore \quad Q_L = \frac{U - L}{\sqrt[3]{\frac{4}{3}\pi(U^3 - L^3)}}$$

8.6) De la ecuación (8.14),

$$\frac{1}{K_{TOTAL}} = \frac{1}{1000} + \frac{1}{300} = 4.333 \times 10^{-3}$$

$$\therefore \quad \boxed{K_{TOTAL} = 230.77 \ \frac{\text{NTM}}{\text{RAD}}}$$

8.16) De la ecuación (8.15),

$$K = \frac{G\pi d^4}{32L} = \frac{(0.33 \times 7.5 \times 10^{10})(\pi)(0.001)^4}{(32)(0.40)} = \boxed{0.006135 \ \frac{\text{NTM}}{\text{RAD}}}$$

¡Esto es muy insustancial, ya que el diámetro es de 1 mm!

CAPÍTULO 9 EJERCICIOS SOBRE CONTROL LINEAL DE LOS MANIPULADORES

9.2) De la ecuación (9.5),

$$s_1 = -\frac{6}{2 \times 2} + \frac{\sqrt{36 - 4 \times 2 \times 4}}{2 \times 2} = -1.5 + 0.5 = -1.0$$
$$s_2 = -1.5 - 0.5 = -2.0$$

$\therefore$
$quad x(t) = c_1 e^{-t} + c_2 e^{-2t}$ y $\dot{x}(t) = -c_1 e^{-t} - 2c_2 e^{-2t}$

$$Att = 0 \quad x(0) = 1 = c_1 + c_2 \qquad (1)$$
$$\dot{x}(0) = 0 = -c_1 - 2c_2 \qquad (2)$$

Si sumamos (1) y (2) obtenemos

$$1 = -c_2$$

por lo que $c_2 = -1$ y $c_1 = 2$.

$$\therefore \boxed{x(t) = 2e^{-t} - e^{-2t}}$$

9.10) Usando la ecuación (8.24) y suponiendo el uso de aluminio, obtenemos

$$K = \frac{(0.333)(2 \times 10^{11})(0.05^4 - 0.04^4)}{(4)(0.50)} = 123{,}000.0$$

Utilizando la información de la figura 9.13, la masa equivalente es $(0.23)(5) = 1.15$ kg, por lo que

$$W_{\text{res}} = \sqrt{k/m} = \sqrt{\frac{123{,}000.0}{1.15}} \cong \boxed{327.04 \ \frac{\text{rad}}{\text{seg}}}$$

Esto es muy alto, por lo que el diseñador probablemente esté equivocado al pensar que esta vibración de vínculo representa la resonancia no modelada más baja.

9.13) Al igual que en el problema 9.12, la rigidez efectiva es $K = 32000$. Ahora, la inercia efectiva es $I = 1 + (0.1)(64) = 7.4$.

$$\therefore \quad W_{\text{res}} = \sqrt{\frac{32000}{7.4}} \cong 65.76 \frac{\text{rad}}{\text{seg}} \cong \boxed{10.47 \text{ Hz}}$$

CAPÍTULO 10 EJERCICIOS SOBRE CONTROL NO LINEAL DE LOS MANIPULADORES

10.2) Si $\tau = \alpha\tau' + \beta$

$$\alpha = 2 \quad \beta = 5\theta\dot{\theta} - 13\dot{\theta}^3 + 5$$

y $\tau' = \ddot{\theta}_D + K_v \dot{e} + K_p e$

en donde $e = \theta_D - \theta$

y

$$K_p = 10$$
$$K_v = 2\sqrt{10}$$

10.10) Sea $f = \alpha f' + \beta$

con $\alpha = 2, \beta = 5x\dot{x} - 12$

y $f' = \ddot{X}_D + k_v\dot{e} + k_p e, \qquad e = X_D - X$

$k_p = 20, k_v = 2\sqrt{20}.$

CAPÍTULO 11 EJERCICIOS SOBRE CONTROL DE FUERZA DE LOS MANIPULADORES

11.2) Las restricciones artificiales para la tarea en cuestión serían

$V_z = -a_1$	$F_x = 0$ $F_y = 0$ $N_x = 0$ $N_y = 0$ $N_z = 0$

en donde a_1 es la velocidad de inserción.

11.4) Use la ecuación (5.105) con las tramas $\{A\}$ y $\{B\}$ invertidas. Primero debe encontrar ${}^B_AT$, así que invierta ${}^A_BT$:

$$ {}^B_AT = \begin{bmatrix} 0.866 & 0.5 & 0 & -8.66 \\ -0.5 & 0.866 & 0 & 5.0 \\ 0 & 0 & 1 & -5.0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Ahora,

$$ {}^BF = {}^B_AR\ {}^AF = [\ 1\ \ 1.73\ \ -3\ ]^T $$

$$ {}^BN = {}^BP_{AORG} \otimes {}^BF + {}^B_AR\ {}^AN = [\ -6.3\ \ -30.9\ \ -15.8\ ]^T $$

$$ \therefore \quad {}^BF = [\ 1.0\ \ 1.73\ \ -3\ \ -6.3\ \ -30.9\ \ -15.8\ ]^T $$

Índice